FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Gesture Recognition for Human-Robot Interaction for Service Robots

Patrick de Sousa

FOR JURY EVALUATION

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Luís Filipe Pinto de Almeida Teixeira

Co-Supervisor: Prof. António José Ribeiro Neves

Company Supervisor: Tiago José Arieira Esteves

June 25, 2017

Abstract

Robots are quickly becoming an intrinsic part of our daily lives and it is becoming important to provide the users a simple and intuitive way to interact with them. In this thesis, we present a Human-Robot Interface for an existing service robot. This robot is mostly addressed to people with reduced mobility on the shopping process in dynamic and crowded environments (*e.g.* supermarkets). This interface was created in order to interact with the robot through the recognition of the START, STOP and PAUSE commands.

Interaction can be performed by two types: verbal and non-verbal. In our approach we decided to work in a non-verbal interaction that will receive the proposed commands via dynamic gestures.

A novel method for hand gesture recognition based on depth information was implemented and tested. The software was developed to be used by a robot equipped with a RGB-D camera. This camera captures images in real time where the robot user's position is obtained. Taking as input the information already processed by the robot, the arm/hand is obtained by a depth based segmentation approach. A Principal Component Analysis is then computed to each object and its center of mass and eigen vectors are calculated in order to extract the hand's tip and orientation. A Kalman Filter is then applied for tracking the hand and get its position through time. Given this information and based on Finite State Machines it is possible to recognize each gesture (START, STOP, PAUSE).

Finally, the proposed gesture recognition approach was tested in a real case scenario with different users obtaining an accuracy around 90%. More specifically, the STOP gesture was recognized with a correct rate of 97.4%, the PAUSE gesture obtained a correct rate of 84.6% and finally the START gesture obtained 87.2%.

Resumo

Os robôs estão rapidamente a tornar-se uma parte intrinseca das nossas vidas e começa a ser importante fornecer aos utilizadores uma forma simples e intuitiva de interagir com eles. Nesta dissertação, apresentámos uma Interface Homem-Robô para um robô de serviço existente, maioritariamente direcionado para pessoas com mobilidade reduzida no processo de compras em ambientes dinâmicos e populosos (ex. supermercados). Esta interface foi criada de forma a reconhecer os comandos START, STOP e PAUSE.

A interacção pode ser realizada de duas formas: verbal e não-verbal. Na nossa abordagem decidimos trabalhar numa interação não-verbal de forma a receber os comandos propostos através de gestos dinâmicos.

Um novo método para o reconhecimento de gestos baseado na informação de profundidade foi implementado e testado. O *software* foi desenvolvido de forma a ser utilizado num robô equipado com uma câmera RGB-D. Esta câmera captura imagens em tempo real em que a posição do utilizador é obtida. Utilizando como entrada a informação já processada pelo robô, o braço/mão é obtido através de uma segmentação seguindo uma abordagem de profundidade. Uma análise de componentes principais é calculada para cada objeto onde é obtido o seu centro de massa e os seus vetores eigen de forma a extrair a ponta da mão e a sua orientação. Um filtro de Kalman é depois aplicado de forma a obter a posição da mão ao longo do tempo. Dada esta informção e com base em máquinas de estado finitas que foram implementadas de forma a descrever os gestos (START, STOP, PAUSE) é realizado o reconhecimento de gestos.

Finalmente, a abordagem proposta de reconhecimento de gestos foi testada num cenário real com diferentes utilizadores onde foi obtida uma precisão à volta dos 90%. Mais especificamente, o gesto STOP foi reconhecido com uma taxa de acerto de 97.4%, o gesto PAUSE obteve uma taxa de acerto de 84.6% e finalmente o gesto START obteve 87.2%.

Acknowledgements

I would like to thank Luís Teixeira and António Neves, my supervisor and co-supervisor respectively, for their academic guidance during my dissertation, their help solving my problems and their comprehension.

Thanks to Tiago Esteves, my supervisor from Follow Inspiration S.A. that helped me when I needed and became a real friend. He guided me not only in my dissertation but in also in my life sharing his wisdom and sense of humor. Thanks to Inês Domingues for helping me in the integration in the company and guiding me in the first steps of this dissertation. I want to thank Follow Inspiration S.A. for the opportunity given and its entire team for their friendship, for making me fell part of the team, for providing me all the best conditions and facilities and for always helping me to accomplish this work.

I want to thank all my friends specially João Correia Pinto and Manuel Camarneiro for always supporting me, with their friendship, dinners and free rides to the company. I owe them a lot. I also want to thank my best friend Cátia Silva that was always present to cheer me up when I needed.

I can't forget to thank all my friends that always accompanied me during these years.

Last but not least I want to thank my family for their support and effort. Without them I would not be able to complete this path.

"Fall seven times and stand up eight."

Japanese Proverb

Contents

List of Figures

List of Tables

Abbreviations and Symbols

HRI	Human-Robot Interaction
GMM	Gaussian Mixture Model
ToF	Time of Flight
CamShift	Continously Adaptive Mean Shift
CONDENSATION	Conditional Density Propagation
DTW	Dynamic Time Warpping
HMM	Hidden Markov Model
FSM	Finite State Machine
CNN	Convulotional Neural Network
LBP	Locally Binary Pattern
FoV	Field of View
ROS	Robot Operating System
IMU	Inertial Measurement Unit
PCA	Principal Component Analysis

Chapter 1

Introduction

1.1 Context

Service Robots have always been a topic in science fiction, since C3-PO from "Star-Wars", or the cleaning robot Wall-e, or even medical assistants like Baymax from "Bighero-6". It became clear that it is a dream for humans to include robots in their daily lives and to create some sort of empathy with the system that helps them in ordinary tasks. Out of science fiction, service robots are becoming real in our lives and the International Federation of Robotics (IFR) defines them as “a robot that performs useful tasks for humans or equipment excluding industrial automation application" [22].

The popularity of these everyday assistants has been increasing exponentially for the last few years and a lot of examples prove it. From taking care of the elders, to cleaning houses (Roomba presented in Figure 1.1) or even helping with shopping (OSHbot presented in Figure 1.2,wGO presented in Figure 1.4).

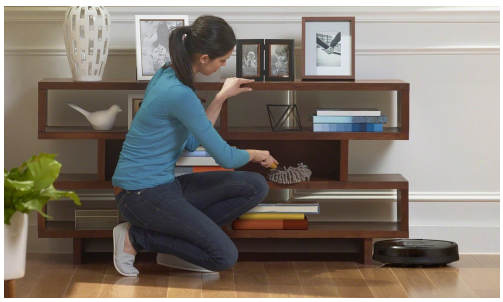


Figure 1.1: Roomba helping in the house keeping [1].

Figure 1.2: OSHbot interacting with a person in a shop [2].

1.2 Motivation

This project is held in a partnership with an innovative Portuguese start-up, Follow Inspiration S.A. (Figure 1.3). This company is responsible for the creation of a pioneer robot designed to

improve the customer's in-store experience by following them while carrying their shopping bags (wGO presented in Figure 1.4).

Figure 1.3: Logo of the company Follow Inspiration S.A.[3].

Figure 1.4: wGO in the shopping process [3].

The current interface of wGO has a screen to show information to the user and a button to start/stop the system. Since the target users of wGO are people with reduced mobility, the company felt the necessity to implement a more user friendly interface. This would allow an easier use and also make the user feel more connected with the wGO.

1.3 Objectives

The main objective of this dissertation is to develop a Human-Robot Interface that will allow the user to communicate with the robot using gestures. By doing that, the robot will be easily controlled by the user, making the shopping experience smooth and interactive.

Using a RGB-D sensor, the system already detects the user and this information allow us to focus on the human gestures. This system is meant to be developed in ROS, a very popular framework for robot software. This is already being used in the wGO development, which will facilitate the system integration. Firstly, the HRI will be tested independently of the system, and later it will be integrated in the wGO and tested in a real scenario. In order to adapt the HRI to all the possible users, we will study and suggest gesture commands, that are easy to perform by low mobility people to be recognized by our system.

1.4 Contribution

This work accomplishes the objectives of this dissertation and contributes to the study of Gesture recognition systems for service robots. The results include:

- A novel approach based on depth image for identifying and tracking the hands of the user.
- Gesture classification based on Finite State Machines.
- Integration of a gesture recognition system in a Service Robot.

The contributions are sumarized in a scientific paper accepted to be published in the VipIMAGE conference with the title "Human-Robot Interaction based on gestures for service robots" [23].

1.5 Thesis outline

The next chapter presents the study of existent service robots and their HRI. It also presents different HRI with focus on non-verbal interaction through gesture recognition.

Chapter 3 introduces the wGO and analyzes the tools and software available on it, specially the ones found useful for our gesture recognition approach. The integration of our gesture recognition module in the wGO is planned.

In Chapter 4 the approach for our gesture recognition is presented and described in detail. It starts by describing the system overview and the decision of the gesture to be performed and recognized. Also in this chapter, and as the main theme, we present the implementation of the approach. The method is tested and the results are presented and analyzed.

Chapter 5 is reserved for the conclusions and future work regarding this project.

Chapter 2

Literature Review

In this chapter, the literature review is presented for this project, talking about some Service Robots and Human Robot Interaction (HRI). In the section of Service Robots, some examples of existent service robots are presented from commercial to research solutions. It is also presented the literature review of HRI with focus on non-verbal interaction via the recognition of dynamic gestures.

2.1 Service Robots

In the last years, the advances in technology allowed service robots to operate in a less structured and more uncontrolled environment. Robots are now performing more complex tasks and interacting with humans in a more natural way. It leads to a faster proliferation of this kind of robots. Service Robots have for long been an object of research but only now are considered an emerging market [24].

2.1.1 Commercial solutions

As commercial products, many have been the service robots helping us in our houses. From helping in the house, keeping it clean, to companion robots, personal assistants, educational or entertainment robots. There have been a huge variety of applications for service robots [24].

One of the first robots to appear in our houses was Roomba from the company iRobot. Roomba is an autonomous vacuum cleaner robot and was presented in 2002 [1].

Figure 2.1: Roomba robot cleaning [1].

Blue Frog Robotics developed a companion robot called BUDDY. BUDDY interacts with people by speech recognition and recognizing the person. It can work as a personal assistant reminding the user of important tasks and events and giving practical information like weather, recipes, etc., play with the user or even showing multimedia content, and many other services [4].

Figure 2.2: BUDDY interacting with a family [4].

In order to support humans in everyday environments, Fraunhofer developed Care-o-bot that is already in its fourth version. Care-o-bot is a mobile robot assistant designed to actively help humans in their daily-life. It can perform a variety of tasks since delivering food and drinks to assist in the kitchen or cleaning. It can also be used in applications outside of our homes like supporting patients in hospitals, to help in restoration delivering orders or performing reception and room service in hotels [5].

Figure 2.3: Cara-o-bot 4 assisting in kitchen [5].

Like Care-o-bot 4, service robots are also used in tasks outside the home environment like hospitals, restaurants, malls, deliveries, etc.

Chinese company Sanbot developed Sanbot S1, a service robot that has many cases of application. It can be used in education to help children, in health-care, to help doctors by monitoring the patient and keep track of the medical records, in retail, in order to greet customers and guide them through store and in security with the help of face recognition to detect strangers [6].

Figure 2.4: Sanbot S1 helping doctors to take care of patients [6].

For security, Cobalt Robotics develops indoor robots to cooperate with human guards in surveillance patrolling offices, museums, schools, and keeping them safe, looking for intruders or anything strange [7].

Figure 2.5: Robot from Cobalt Robotics performing the surveillance of an office [7].

Lowe's Innovation Labs created OSHbot. This robot is used in retail, customers can tell what they are looking for or just hold up an item for it to scan. Then OSHbot can guide them to the product they are looking for [2].

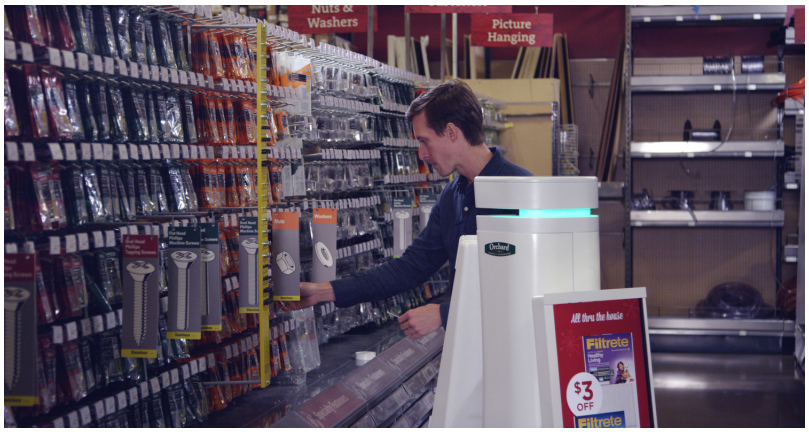


Figure 2.6: OSHbot helping in the shopping process [2].

2.1.2 Research Projects

From research, many have been the projects that allowed to bring new technologies to service robots. A league called RobotCup@Home, part of the RoboCup initiative, was created with the aim to develop service and assistive robot technology with relevance for future domestic applications. Some domains explored in this competition are interaction and cooperation with humans, navigation and mapping in dynamic environments, object recognition and manipulation, behavior integration. The competition is performed in a real world scenario and some of the factors evaluated are the human robot interaction, the social relevance, the time to perform the task, the easy set up and low cost [25].

One of the most relevant participants of the RobotCup@Home is the project AMIGO (Autonomous Mate for IntelliGent Operations) developed by Eindhoven University of Technology with the aim of allowing older people to be independent in their houses [8].

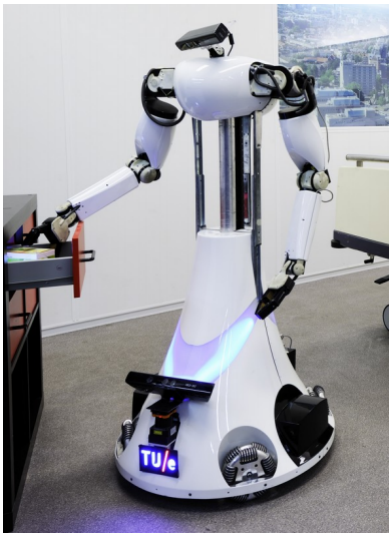


Figure 2.7: AMIGO robot from the Eindhoven University of Technology [8].

The MOnarCH project (Multi-Robot Cognitive Systems Operating in Hospital), funded by the European Commission, developed a robot working on the integration of robots in social spaces. The robot developed was used in the pediatric infirmary in the Portuguese Oncology Institute at Lisbon (IPOL), Portugal with the aim to entertain and educate children, staff and visitors [26].

Figure 2.8: Monarch with a child at IPO. Picture by Miguel A. Lopes/Lusa.

One of the best examples is the robot from the European research project called SPENCER. With the motivation of the increasing number of robots sharing space with people the aim of this project was to break new ground for cognitive systems in populated environments. The technologies developed were integrated in a robot platform whose aim was to guide people at the airport [9].

Figure 2.9: SPENCER interacting with two persons in Airport [9].

2.2 Human-Robot Interaction

HRI studies how humans and robots interact with each other in the more effective and natural way [27].

Nowadays, with robots entering in our daily lives, it is becoming important to provide the users a simple and intuitive way to interact with them. Human-Robot interaction has already proved to be a major field in robotics with an increasingly investment in more rich and innovative kinds of interaction. Most of those interactions are based on verbal or on non-verbal interaction [28].

Verbal interaction have been used for a long time as humans recur to voice to communicate and interact between each other. Thus, this is also used by robots to assure a natural interface between humans and machines [29]. Speech recognition allows the robot to perceive voice commands and take actions depending on the received instructions. There are a few libraries available to implement such solutions [30].

Non-verbal Interaction involves all interaction except the speech and is commonly used by humans, specially facial expressions and gestures.

2.2.1 Gesture recognition

Gestures are a way of non-verbal communication and can be made from any bodily motion or state but they are usually made from the face or hand [31]. Gestures can be divided into two types according to their movement along time: **static** or **dynamic**. Static gestures do not change with time, they are described by the pose/posture in a single instant. Dynamic gestures change the posture across time and the gestures are described by its movement [32]. Gesture Recognition is the process of identifying the gesture performed by a user and usually has the aim of interpret certain commands [33]. We divided the gesture recognition process in four important parts, **Data Acquisition** where the information from the ambient is acquired, **Segmentation** where the features

necessary to perform the gesture recognition are extracted, **Tracking** where the hand is tracked across time and **Classification** where the gestures are modelled and recognized.

2.2.1.1 Data acquisition

In order to perform the gesture recognition it is necessary to acquire the data from the user and for that, the approach can be from one of these types: motion sensor-based or vision based.

Motion sensors, like gyroscopes, accelerometers, bend sensors and others, allow the acquisition of movement from the desired joints. In order to detect the movement of some skeleton joints to perform a gesture recognition system, Alavi et al. attached five wireless IMUs (inertial measurement units) to their arms and upper body. IMUs have accelerometers that give the acceleration and gyroscopes that have angular velocity as output [34]. Usually, to detect the motions from the hands, data gloves are used. Adnan et al. developed a low cost dataglove using bending sensors for the index and middle fingers for measuring its bending for various virtual interaction [35]. Kim et al. created a data glove with three tri-axis accelerometer sensors, one for the hand palm, another for the thumb and the other for the other middle finger, the information is sent via Bluetooth to a PC where a 3D digital hand model for hand motion tracking and recognition is implemented [10].

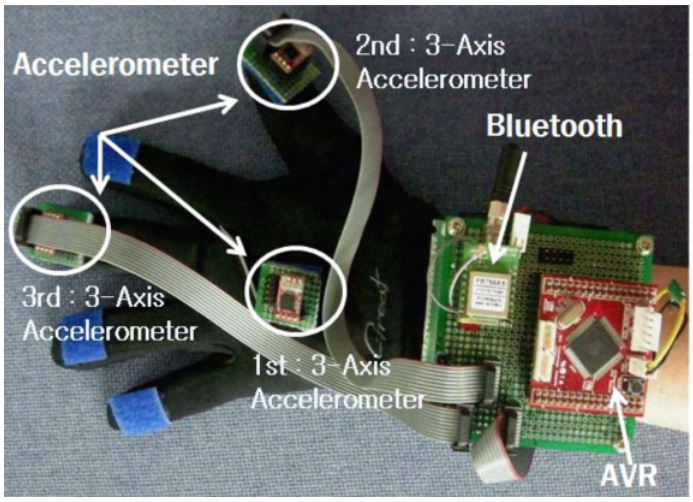


Figure 2.10: Data glove implemented by Kim et al. [10].

Instead of using a glove and since smartphones usually have accelerometers and gyroscopes, Gupta et al. used a smartphone to recognize gestures in order to control smart appliances [36]. Since this kind of sensor is attached to the user, the data acquired is not affected by its surroundings and it provides a better coverage of the movements since vision-based can suffer occlusions or variations of the light [10]. As a drawback, these approaches are intrusive for the user and do not allow a very natural interaction [37].

Vision-based solutions are user independent and have emerged to give a most natural experience to the user. RGB cameras were the first used to acquire data, Argyros et al. [11] and Cristina Manresa et al. [38] used RGB cameras to acquire the hand in a 2D plan. To acquire the information in a 3D space, that allows more complex motion gestures, depth cameras like Stereo, Time

of Flight (ToF), Leap Motion or Structured Light cameras have emerged. Park et. al. [39] and Cerlinca et. al. [13] both used depth information from a structured light camera (Kinect) to track the position of the hand in space.

Figure 2.11: Structured Light camera Kinect v2 from Microsoft.

The vision-based approach, unlike the inertial motion-based approach, gives hand features by performing hand/arm segmentation and extracting the desired features from it to recognize the gestures.

2.2.1.2 Segmentation

In order to perform hand/arm segmentation, the most popular method is to do a segmentation based on skin-color. Argyros et al. proposed a method for detecting skin-colored objects using a Bayesian classifier with a small set of training with an on-line adaptation of skin-color probabilities to cope with illumination changes [11].

Figure 2.12: Skin color segmentation [11].

In order to perform a pointing gesture recognition, Park et al. developed a system where after detecting the face of the user, the hands were detected assuming a similar color of the face [17]. To minimize some error that can occur because of the illumination changes, models of the variation of the skin color with the light are used. Liu et al. recognizes hand gestures with a stereo camera, by performing an online color calibration at the beginning of the process where the user places his or her hand in a region of interest and a Gaussian Mixture Model (GMM) is trained to cope with

the variation of light. The segmentation is then performed recurring to the GMM with a region of interest specified by a tracking module [12].

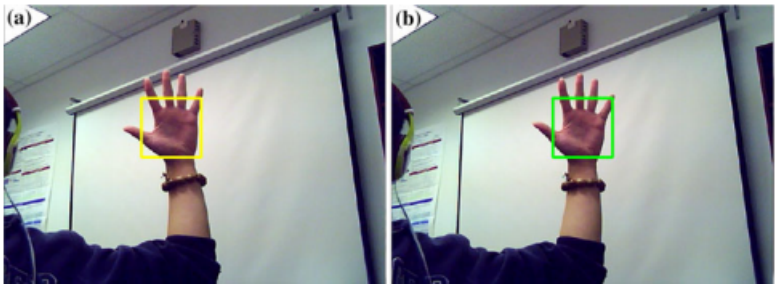


Figure 2.13: Online color calibration: a) left camera calibration box, b) right camera calibration box [12].

Rodriguez et al. detects hands in order to perform a 3D gesture interaction with a RGB camera. The author proposed an approach based on the Viola-Jones detector [40] using an AdaBoost algorithm to select a set of Haar-like features. This approach usually is used for face detection but it can be used to detect another objects. The distance of the hand from the camera was determined by the size of the hand [41].

This kind of approaches based on skin color is efficient but it has the problem that the user can not wear any kind of gloves and it can not appear skin colored objects in the background and some are susceptible to light variation.

A common method used for hand detection is to perform a simple segmentation applying a threshold based on distance, recurring to cameras with depth information. The distance used can be selected regarding to another part of the user. A real-time 3D hand detection was implemented by Cerlinca et al. assuming that the hands are always closer to the acquisition sensor than the head. As so, a face detection was implemented and its location on the depth image was used to obtain its distance to the sensor and then this distance used as a threshold value to obtain the arms. A region growing algorithm was used to refine the hands [13].

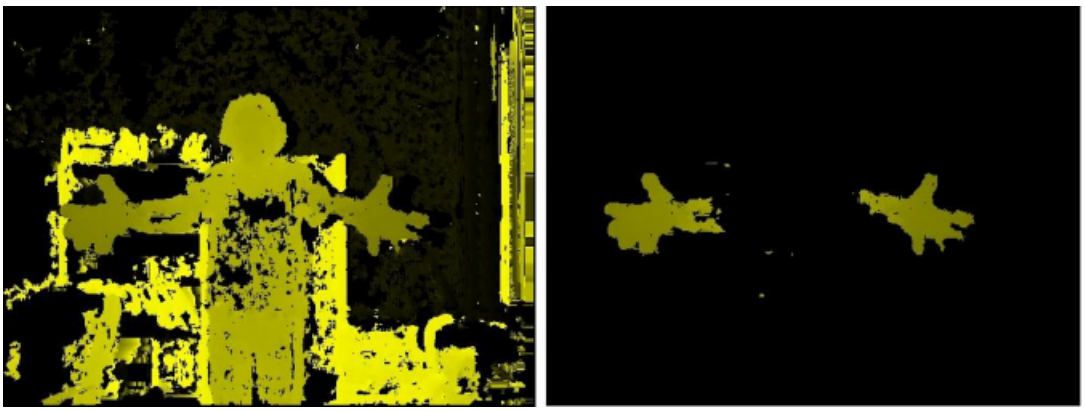


Figure 2.14: Depth map filtering with distance threshold [13].

In order to implement a real-time hand tracking, Chen et al. used a region growing technique with the seed point on the estimated center of the hand based on the previous frame. The estimation of the first seed was obtained by a hand click detection method in order to start the tacking [14].

Figure 2.15: Hand region and center obtained by Chen et al. [14].

Sometimes both skin-color and depth information are used. Bergh et al. used a Time of Flight (ToF) and an RGB camera for a real-time 3D hand gesture recognition, the face was detected and the distance from it to the camera was measured. Based on this distance, a threshold was applied to the depth image to discard background objects. The remaining pixels, together with skin color detection based on a GMM trained with the variations of illumination, were used to detect the hands [15].

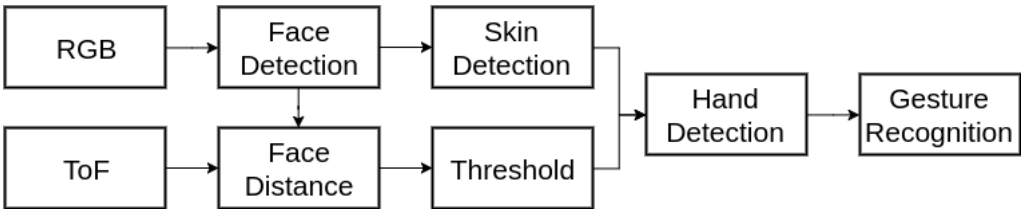


Figure 2.16: System overview with the RGB and ToF camera used proposed by Bergh et al. [15].

Park et al. proposed a different approach to detect the hand for a hand tracking implementation where the hands were detected by using motion clusters and predefined wave motion [39].

With the emergence of skeleton tracking algorithms like OpenNI with NITE [42] and Kinect SDK [43], it was possible to obtain the skeleton of the user with the information of the most important joints including arms and hands. Bellmore et al. used NITE to obtain the pose of the observer to interact with an interactive display. This approach requires a calibration pose to initialize body tracking [44]. Ghotkar et al. presented an Indian Sign Language recognition using the Kinect SDK in order to obtain the joints of the user [16].

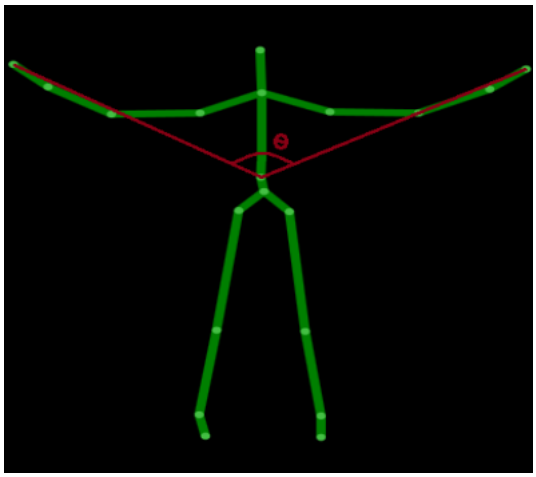


Figure 2.17: Skeleton segmentation and extraction of the angle beetwen the arms by Ghotkar et al. [16].

2.2.1.3 Tracking

With the resultant segmentation and the hand/arm identified and its desired features obtained, several algorithms can be used to perform the tracking of the desired object.

Mean-Shift algorithm is a non parametric technique used to find modes of density of a distribuition by climbing the gradient of the probability of the distribution. And it is efficient to tracking objects whose appearance is defined by histograms [45]. Chen et. al. used MeanShift algorithm to track the hand identifying the center of the palm [14]. MeanShift does not deal with the variation of size of the tracking object. **Continuously Adaptive Mean Shift (CamShift)** algorithm is similar to the MeanShift but it also adjusts the search window size and rotation that it is useful in hand tracking since the hands can appear near or far from the camera changing its size [46]. To track the hand, Liu et. al. implemented a CamShift algorithm based on the Hue component choosing the window of Camshift as the region of interest and its center as the seed point [12].

Figure 2.18: Sample Hue histogram used for CamShift hand tracking by Liu et al. [16].

Kalman filter provides an efficient computational (recursive) means to estimate the state of a process from noisy measurements and its previous state, it has been widely used in a variety of

research fields and real application areas providing many advantages in digital computing [47]. A Kalman filter was used by Park et al. to continuously track the hand's location first obtained by motion clusters and predefined wave motion [39].

Figure 2.19: Hand tracking using Kalman Filter by Park et al. [16].

In order to track the hand after being detected, Rodriguez et al. implemented a Kalman filter that allowed to predict the next position of the hand helping in the elimination of false positives [41].

Since Kalman filter is based on Gaussian densities it cannot represent simultaneous alternative hypotheses for its tracking [48]. **Conditional Density Propagation (CONDENSATION)** is a particle filtering algorithm that uses a entire probability distribution in order to track an object's state [49]. To perform pointing gesture recognition for mobile robots, Park et al. used a particle filter was used to track both hands and face in order to determine at where the user is pointing [17].

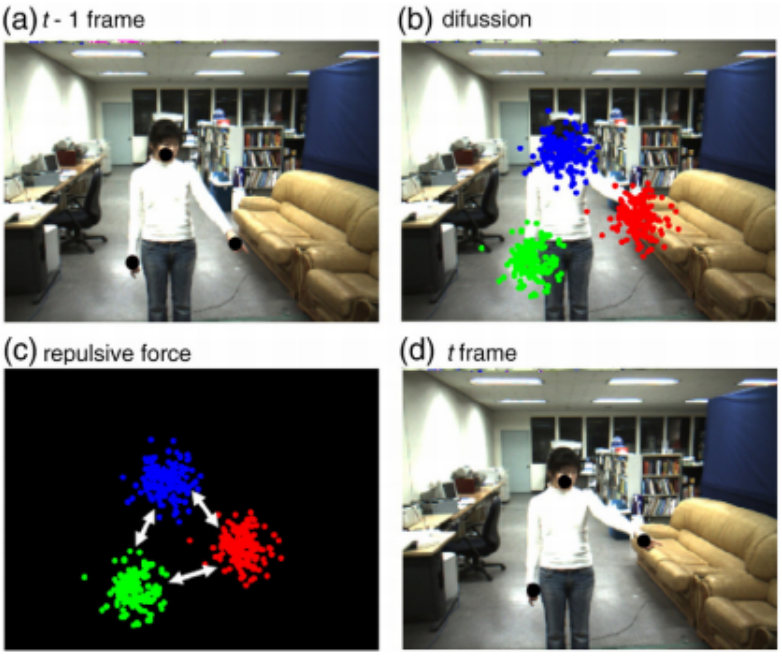


Figure 2.20: Face and hand tracking using 3D particle filters proposed by Park et al. [17].

2.2.1.4 Classification

In order to identify the gesture across time, some classifiers algorithms are used.

Dynamic Time Warping (DTW) is used to find the alignment of two signals. It computes from two signals, the distance between its points [18]. To identify seven dynamic gestures, after tracking the hand by a Cam-Shift algorithm, Liu et. al. proposed a DTW algorithm to recognize gesture by comparing them with a series of prerecorded gestures, obtaining an accuracy of 92.4 % [12].

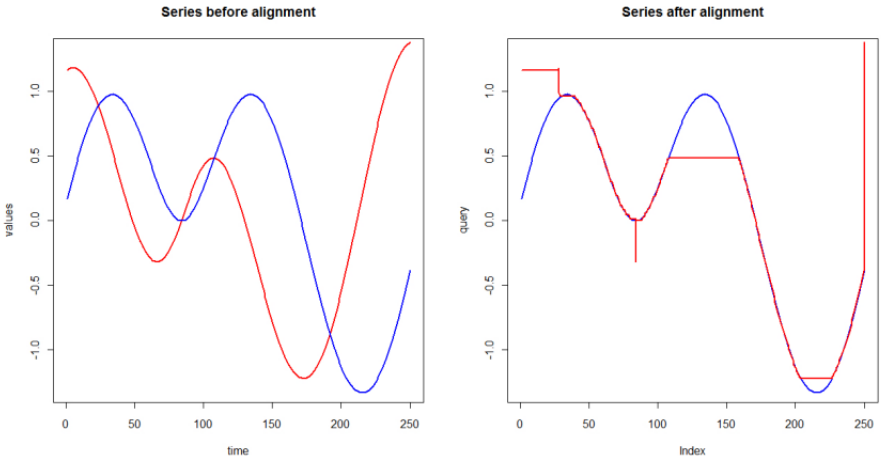


Figure 2.21: The optimal alignment of the two time series [18].

Hidden Markov Models (HMM) are a tool for modelling time series of data representing probability distributions over sequences of observations [50] [51]. Yang et al. applied an HMM to identify eight gestures to control a music application in order to adjust the volume and change the music [52]. To be able to identify 20 dynamic sign of the Indian Sign Language, Ghotkar identified them with an HMM with an accuracy of 89.25%.

For simple and easy model gestures, **Finite State Machines (FSM)** can be applied by modelling a gesture as an ordered sequence of states. Ramey et al. used a finite state machine to classify a simple gesture of waving hand varying the x coordinate to left and right, in order to integrate with a social robot [19].

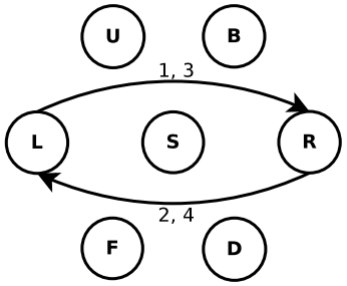


Figure 2.22: Example of the used FSM by Ramey et al.. The state represent "hello" gesture that involves waving the hand. The numbers indicate the state changing sequence [19].

The use of **Neural Networks** is growing in the last years and many of them are being used for gesture recognition. Molchanov et al. proposed a Convulotional Neural Network (CNN) to detect gestures from different inputs since depth, color or stereo-IR sensors achieving a 83.8% accuracy [53].

Next, we start by presenting the robot (wGO) which we will work with in order to develop an HRI for its control.

Chapter 3

wGO Case-Study

In this work, the goal was to create a Human Robot Interface for an existent robot in the market. For this, it is important to study the tools and software available that could be useful and the restrictions that we would have to deal with.

The robot used in this project was the wGO. The wGO is the main product of the company Follow Inspiration S.A. and the aim of this service robot is to improve the customer's in-store experience. This robot was thought and designed to follow its user while carrying their shopping bags across the supermarket. The wGo significantly helps people with reduced mobility even though, any other person is also allowed to use it.

In order for the system to work, it should perform a first valid user recognition. After this quick process it makes use of a sensor set composed by a laser, system of cameras and a set of sonar sensors to track the user, follow him and avoid obstacles which may appear in the way.

Figure 3.1: Render of wGO [3].

3.1 Interaction with the user

In the current version of the wGO, all the interaction with the user is performed by a button that is pressed to start and stop the robot. A sound warning is also played when the robot loses track of the user. There is also a screen which shows some useful information for the user to perceive the current status of the robot or for the development team to check for hardware problems.

3.2 User tracking

The visual tracking system, which has main preponderance in this work, does not require the user to wear any kind of marker or special clothes. The data acquisition for this process is done using a RGB-D camera which captures depth and RGB images. This RGB-D camera is a structured light camera, an Orbbec Astra from the company Orbbec.

Figure 3.2: Orbbec Camera used in wGO[20]

Structured light cameras work by projecting an Infrared Pattern to the environment, obtaining the distance (depth) and merging it with the RGB data [54].

From the camera two images are obtained, an RGB image (Figure 3.3.a) and a Depth image (Figure 3.3.b). These images are matrices of pixels with size 640*480 where each pixel in the position x_i y_i has corresponding RGB values (Figure 3.5) and a depth value in the depth image (Figure 3.4) representing the distance from the camera to the object depicted there.

(a) (b)

Figure 3.3: Data acquisition: a) RGB image with user detection (yellow rectangle) and face detection (pink circle). b) Depth image with user detection.

The current software on wGO, makes use of the RGB-D information, to compute the identification of the user on the image. It obtains a bounding box (x,y, width and height) representing the position of the user on the images captured (Figures 3.3 a and b). A face detector algorithm is also implemented to recognize the front part of the user. For that, a Locally Binary Patterns (LBP) cascade classifier is used, since it is faster than other approaches like Haar and it is very important for this system to work in real time [55]. The depth and RGB images acquired by the robot with the user and face detection are presented in Figures 3.3 a and b.

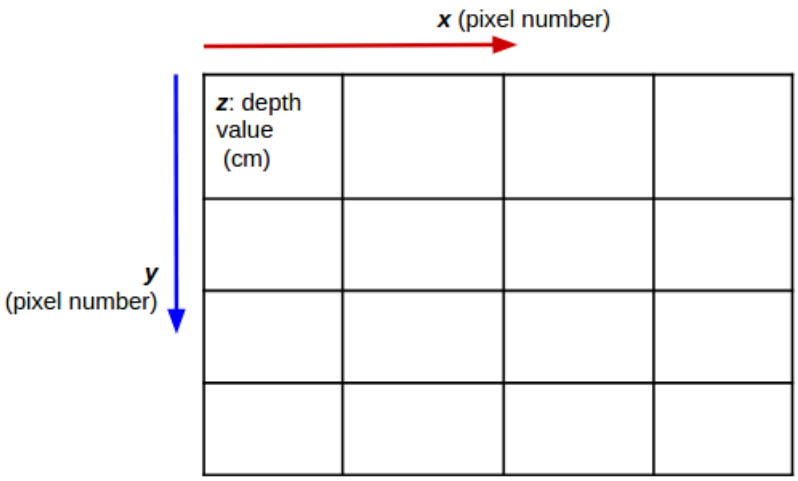


Figure 3.4: Representation of the depth image and the coordinate system (x,y) and the pixel value as the distance from the camera to the object.

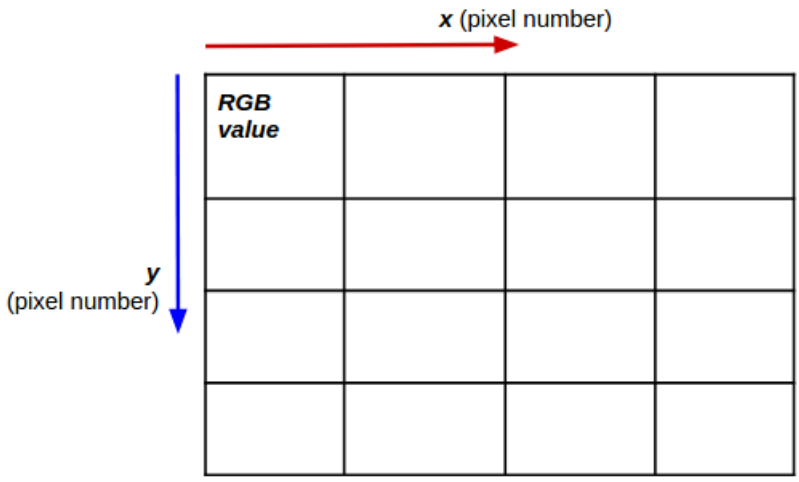


Figure 3.5: Representation of the RGB image and the coordinate system (x,y) and the color of the object.

As we can see in the Figures 3.3 a and b, the Field Of View (FOV) of the cameras with the distance of usage of the wGO restricts the area of the user observed, a fact that is relevant to define our approach. Besides that, the minimum depth detected is 30cm which also affects the possible positions of hands being tracked.

3.3 Robot Operating System

The robot software is developed recurring to the Robot Operating System (ROS). ROS is an Open-Source framework for developing robot software. It is an agglomeration of tools, libraries and conventions with the intent of helping in the creation of robot software that can be written either in Python or C++ [56].

Figure 3.6: ROS logo [21]

The major concepts to implement a ROS system are nodes, messages, topics and services. A ROS system is composed by a number of running independent nodes similar to software modules, communicating between each other by messages. It follows a publisher/subscriber approach where each node may either subscribe a certain topic, enabling it to access data transferred through the messages in that topic, or publish a certain topic with relevant information for any other nodes. Nodes can also communicate by services where a request is sent for a certain node and then

a response is expected [56]. A scheme of the ROS system with the nodes and how they can communicate is presented in figure 3.7.

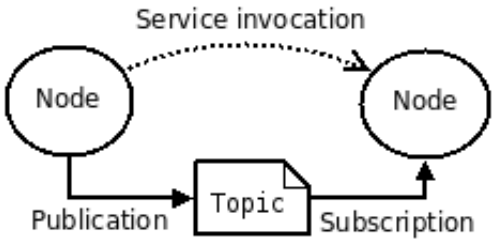


Figure 3.7: Scheme of the ROS system with the nodes and how they can communicate via topics and services [21].

All data from the wGO mentioned in Section 3.2 is already being published in four different topics that can be subscribed by our approach.

3.4 Integration

As soon as the gesture recognition module is completed, it is necessary to send a command to the robot every time a gesture is recognized.

For an easy integration, without changing the wGO current software, we plan to simulate the button press in each corresponding gesture recognition. The button calls two different services to signal the robot when it is pressed or released. To correctly reproduce this action we call the release service immediately after the press signal. Since the information of the button is the same for any state, and it is not able to distinguish the command, we subscribe the topic with the current state of the wGO in order to make sure the command is valid. For instance we only send the start command if the robot is in standby mode. As wGO is not yet prepared to assume a paused state (it can either be stopped or following) we test our solution by sending a stop command when the paused gesture is performed. The interaction of wGO with the Gesture Recognition Module is described in Figure 3.8.

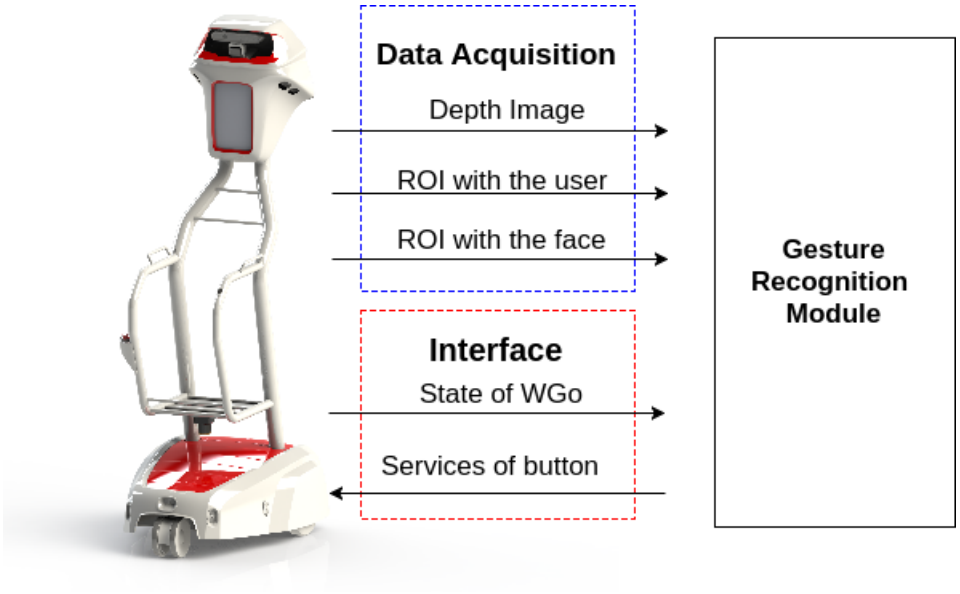


Figure 3.8: Interaction of wGO with the Gesture Recognition Module.

In order to escalate the functions of the robot in the future, we implement a topic (`/Wiigo_HRI/Cmd`) to broadcast the signals perceived by the HRI module.

A set of possible values is defined as follows: 1 for the START, 2 for the STOP and 3 for the PAUSE command.

The wGO will behave depending on its current state and the receiving the command message .

It might also be useful to feedback about the position of the hand and its movement on the graphical interface. To do so, a message is created to detail all the screen node required information (`/Wiigo_HRI/Hand_position`).

`Hand_position`

```
bool left_hand
int8 left_handposition_x
int8 left_handposition_y
int8 left_handposition_z
int8 left_orientation
bool right_hand
int8 right_handposition_x
int8 right_handposition_y
int8 right_handposition_z
```

```
int8 right_orientation
```

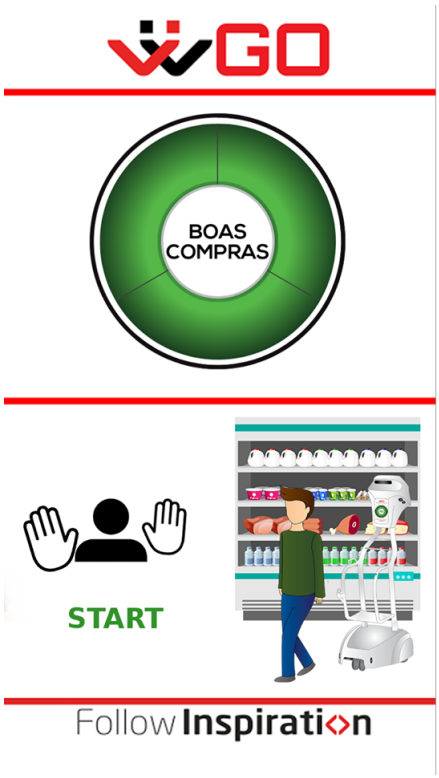


Figure 3.9: Proposed wGO interface with the position of the hands and the recognized command.

The interface proposed for the user to have visual feedback of his commands is presented in the Figure 3.9. By showing this information, the user knows if the hand is being correctly detected, working as a feedback that makes the user to do the gesture in a correct way.

Chapter 4

Gestures recognition

In this chapter we present our gesture recognition approach based on the work presented in the paper "Human-Robot Interaction based on gestures for service robots" accepted to be published in the VipIMAGE conference [23]. We start by describing our system overview and deciding the gestures for the commands. We also describe all the phases of the implementation from data acquisition and segmentation, for the identification of the hand and its tracking, and ending with the classification of the gestures. Our approach is tested and the results are presented and analyzed.

4.1 System Overview

Our system is divided in four main phases (figure 4.1). In the first, the data acquisition from the robot is performed. The current robot's software captures the depth and RGB images and computes the user's position. In the segmentation phase, the user is extracted from the background and then the arms are segmented. In the Hand Detection phase, the position and orientation of the hand is obtained. Given this data, in the Tracking phase a Kalman filter is applied to track and filter the hand position so we can identify the gesture using Finite State Machines in the Gesture Classification part.

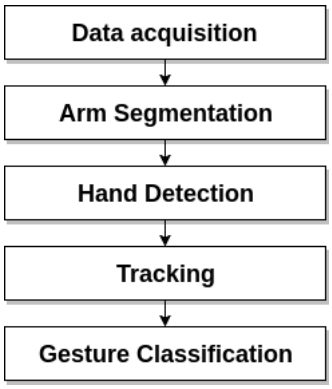
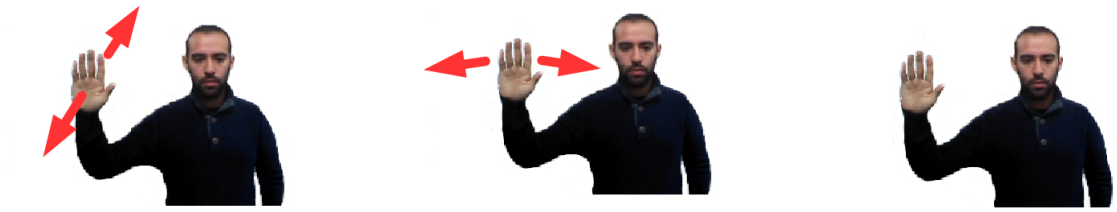


Figure 4.1: System overview of the proposed solution.

4.2 Gestures parameterization

In order to start our Human-Robot Interface, it was important to discuss what type of gestures would be best suitable for the target end-user. Since the main target is people with reduced mobility, it is important to minimize the constraints on their use, due to possible physical limitations. Since the user will not receive any training to operate it, the gestures have to be natural and simple so that he can learn them and do not forget it until the next utilization. Given those facts, we reached to the conclusion that the gesture should be performed by one single arm due to people using mobility aids. Besides that, due to possible low sensitivity in the movements of hands it was better to get the arm position for the gesture instead of the hand pose.



(a) (b) (c)

Figure 4.2: Gestures parameterization: a) `START` gesture where the hand movement is forward and backward towards the robot. b) `PAUSE` gesture formed by the lateral movement of the hand. c) `STOP` Gesture where the hand does not change position.

The chosen gestures, START (where the robot correctly initiate its process), STOP (shuts down the robot processes) and PAUSE (puts the robot in pause mode but still working), are represented in figure 4.2. For those gestures the only information necessary are the coordinates of the the hand (x,y,z) and the arm's orientation.

4.3 Data acquisition

The wGO already computes some information that can be useful for our approach such as the depth acquired by the RGB-D camera, the computed user location and face detection.

The bounding box of the user allows us to focus on the user and reduce the noise of the scene, removing other people or objects standing next to him.

Since gestures have to be made facing the robot, the information of the face detection is important to check this condition. The position of the head is also useful to restrict the gestures to a certain area, minimizing the interference of outside factors or even misunderstandings. A representation of the important topics subscribed are presented in Figure 4.3.

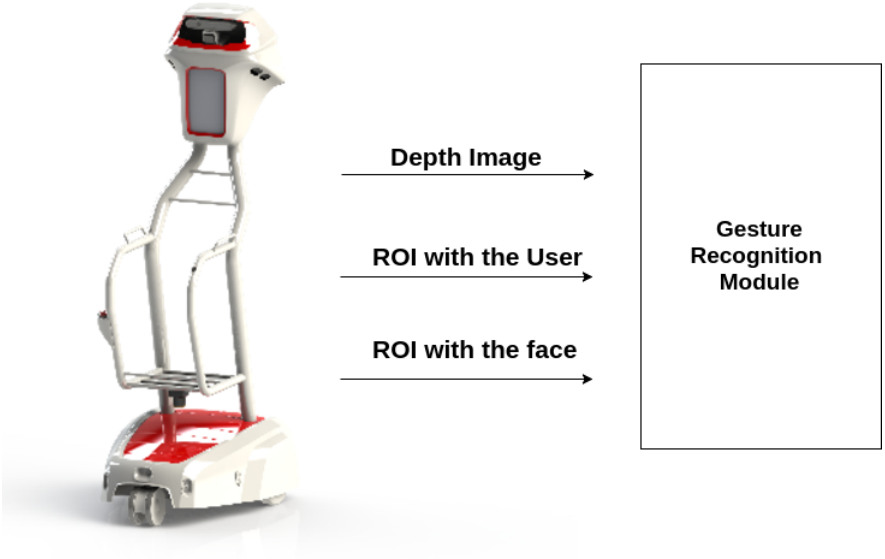


Figure 4.3: Representation of the topics subscribed by the gesture recognition module.

4.4 Segmentation

With the relevant data acquired for gesture recognition, it is necessary to perform segmentation of the desired data in order to extract the user from the background and then extract the arms from the user. The segmentation process is represented in Figure 4.4.

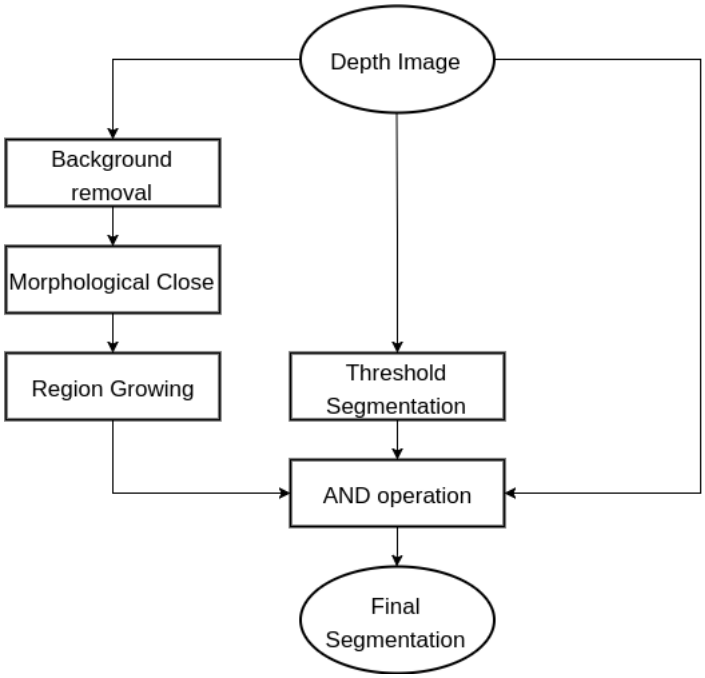


Figure 4.4: Phases of the Segmentation process.

Since the system will be used in a non-controlled environment it is necessary to ensure that only the user will interact with the robot. As we can observe in figure 4.5 the person next to the user is appearing in the image interfering in the segmentation result.

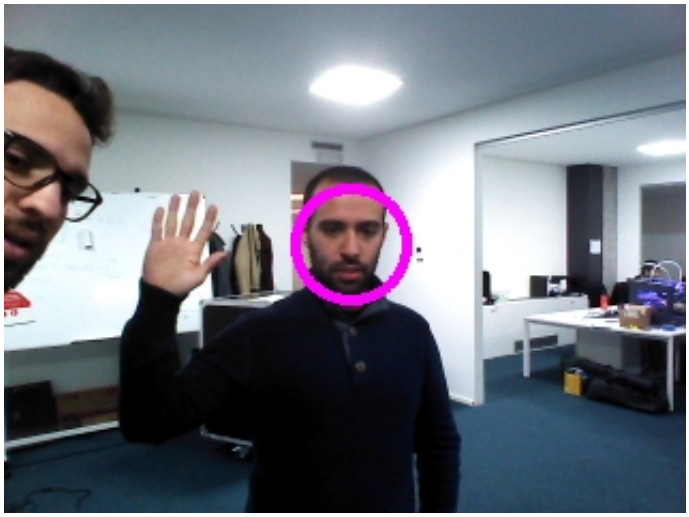


Figure 4.5: RGB image received from the robot with the face detection of the user.

To separate the user from the background (Background removal step in Figure 4.4), we apply an histogram approach to find out the distance from the robot to the user's chest. For this, we consider the user's location information given by the robot and. by focusing on the user's chest. We select the area of the chest (red square in Figure 4.6) by using the limits of the face detection.

Figure 4.6: RGB image representing with a yellow square the identification from the robot, identification and with the red square the region of interest to calculate the threshold value.

Then, the region of interest is extracted from the depth image (red square in Figure and 4.7) from where we compute the histogram (Figure 4.8). We assume that the mode of the histogram is

the user distance to the camera and so it is used as a threshold value.

Figure 4.7: Depth image correspondent to the image 4.6 with the red square representing the region of interest of depth values that will be used to calculate the threshold value to use in further operations.

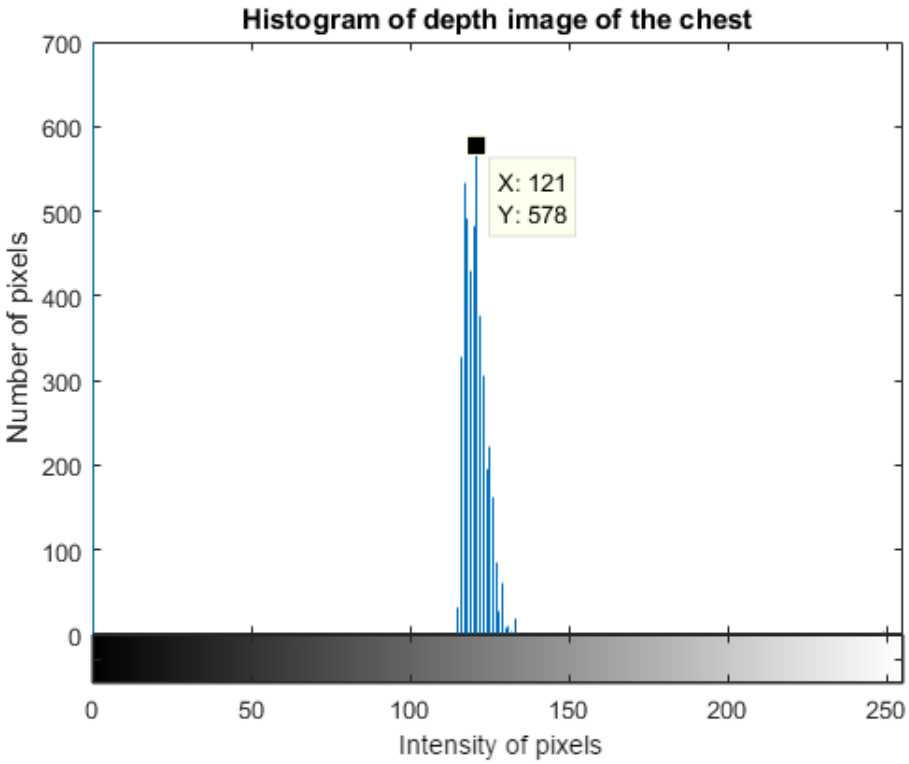


Figure 4.8: Histogram of the pixel values from the region of the interest. The higher value represents the distance of the user from the RGB-D camera.

With the threshold value calculated by the histogram approach we add 15cm to the calculated threshold to ensure that we segment the user totally and the background is removed (Figure 4.9.b).

Since the resultant image is noisy, a morphological close operation was applied in order to reduce the noise (Morphological Close step in Figure 4.4). A close operation is the combination of other two morphological operations, a dilation followed by an erosion [57]. The close operation smooths sections of contours and it generally fuses breaks and thin gulfs, eliminates holes, and fills gaps in the contour. A morphological operation is given by the relation of the image with a structuring element [57]. In this case our structuring element was a circle with 5 pixels of diameter (Figure 4.9.c).

As the user may not be the only object present in the scene close to the camera, in order to focus only on the user, we applied a region growing algorithm (Region Growing step in Figure 4.4. Region growing is a procedure that groups pixels or subregions into larger regions based on predefined criteria. This approach starts by defining a set of "seed" points and from these regions are grown by aggregating the neighbor pixels that have properties similar to the seed (in this case the depth value)[57].In our work the seed was selected as the center pixel of the user (based on the bounding box sent by the robot). The result is the segmented user segmented in a binary image (figure 4.9.d).

In order to isolate the arms from the body another image was obtained by applying another threshold to the original image (Threshold Segmentation step in Figure 4.4). Using the previous value calculated from the histogram it is applied a threshold of this value subtracting 4 cm to obtain an image only with the objects in front of the user (Figure 4.9.e) . Then, we apply an interception (AND operation step in Figure 4.4) between the image after the close operation (Figure 4.9.c) with the region growing mask (Figure 4.9.d) and the image of the threshold ahead of the user (Figure 4.9.e) to obtain only the regions of the user near the camera that we assume as a possible arm of the user. In the original image (Figure 4.9.a) the user and another person appears in the scene with another background noise, with this operation the other user is eliminated from the scene and only the arm of the user is segmented.

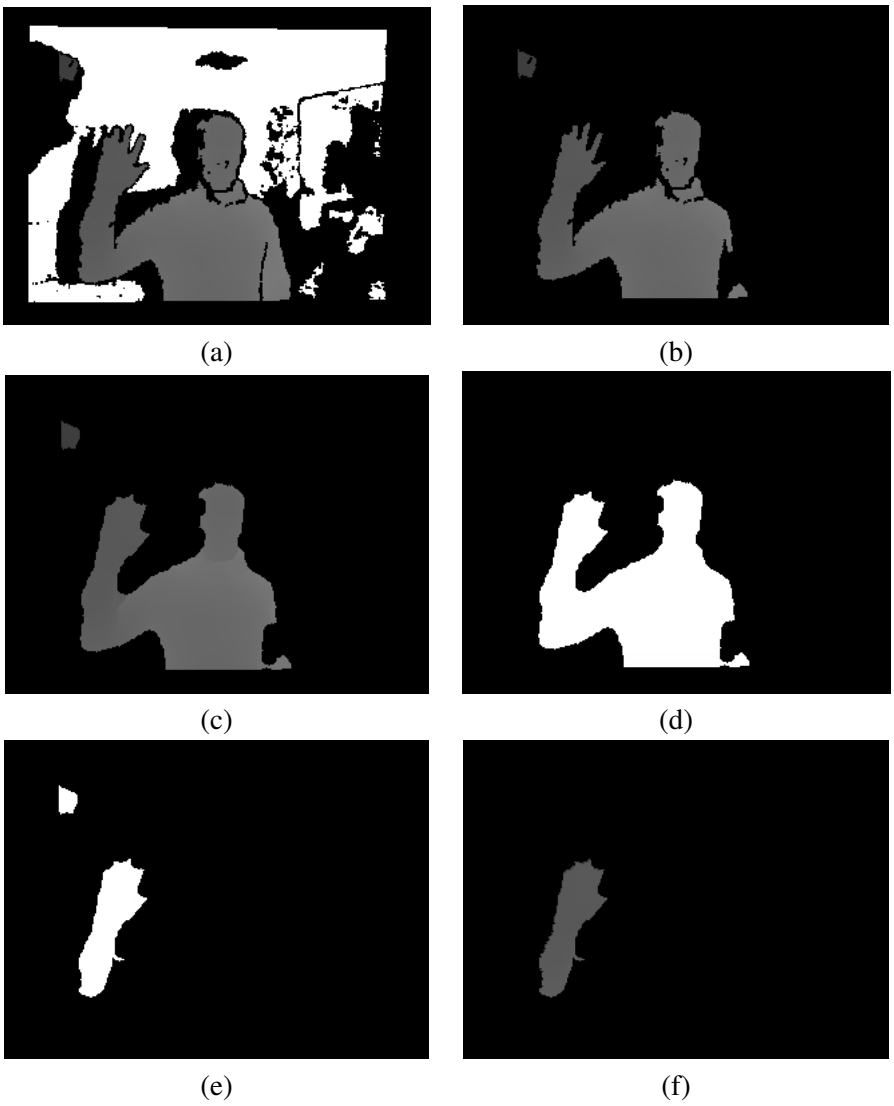


Figure 4.9: User image segmentation: a) Depth image received from the robot. b) Depth image after a threshold to perform the background segmentation. c) Morphological close operation on the threshold image. d) Region growing result with the seed on the center of the user. e) Original depth image after another threshold to extract what is in front of the user. f) Interception of the images c, d and e resulting in the parts of the user in front of him.

4.5 Identification and validation

Segmentation retrieves the most important regions on the image, and then it is necessary to separate them as distinct objects and find the position of the hands in order to detect gestures.

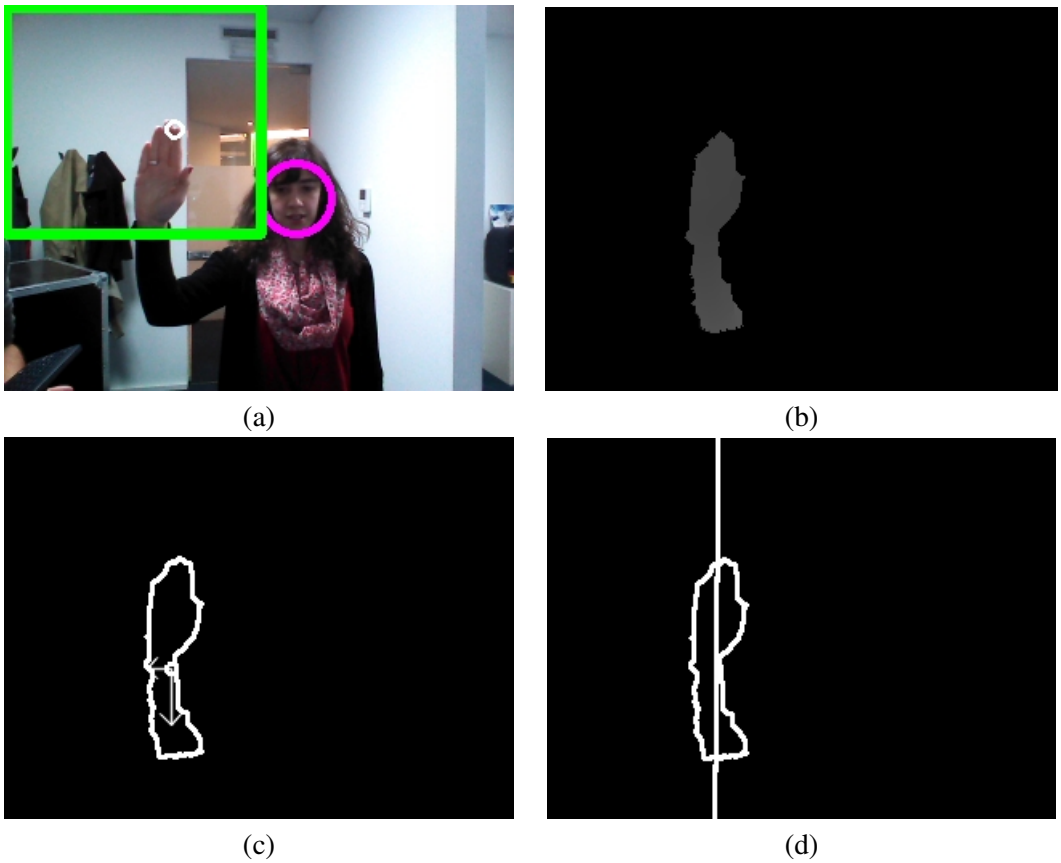


Figure 4.10: Features extraction: a) RGB image with the identification of the hand's tip inside the region of interest (green rectangle). b) Segmentation of the hand and arm. c) Silhouette of the hand and arm with the center of mass and the eigen vectors represented. d) Line obtained by the center of mass and the longer eigen vector crossing the silhouette.

We start by labelling the components and then we perform a Principle Component Analysis (PCA) [58] to each labelled object, which allows the algorithm to understand for each object how its data is distributed across the image, retrieving its center of mass and eigen vectors. The orientation of eigen vectors is considered to draw a line which passes through the center of mass and intersects the silhouette of the object in two different points. We also normalize the orientation's angle to be sure it is pointing to the upper part of the image, assuming that the gesture has to be made with the hand in the upper position of the arm. Thus we can guarantee that the tip of the hand will be the upper intersection point of the line with the object contour.

Gestures presented in section 4.2 are validated using the face position obtained in 4.3 , considering only those which are made on the lateral parts of the face and above the chain, given by the bottom part of the rectangle which defines user's face.

4.6 Tracking

Given the hand tip's position, we are able to track the hand's position across time to be able to identify the gestures performed by the user.

4.6.1 Kalman Filter

As seen in Chapter 2 of Literature Review, the Kalman filter is used for tracking. Since it only depends on the previous estimation and the new measurement, it requires low computing resources and has proven its efficiency. The main goal of the Kalman filter is to estimate the state of a dynamic system from (noisy) measurements and its previous state by a form of feedback control.

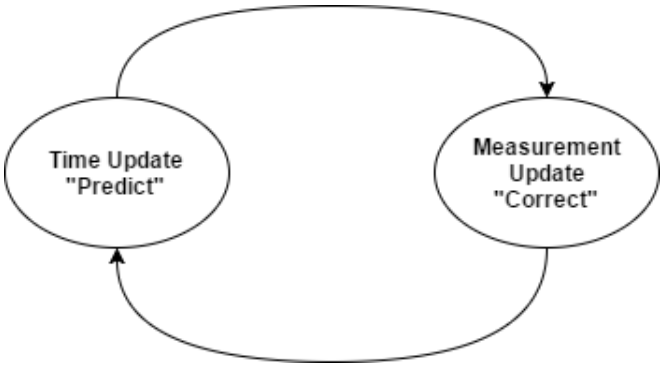


Figure 4.11: The Kalman filter cycle with the two calculation phases (time update and measurement update) and their equations.

The Kalman filter works by creating a loop represented in Figure 4.11, where it first predicts the state of the system (time update phase), and then it corrects the initial prediction with a given new noisy measurement (measurement update). It is assumed that the system is linear and the probability density function follows a Gaussian distribution in each state [39][47].

The equations for the Kalman filter can be divided in two groups:**time update** and **measurement update** equations.

The current state and error covariance estimates are computed by the time update equations in order to obtain *a priori* estimates for the next step. Being responsible for the feedback, recurring to the *a priori estimate* obtained with the time update equations and a new measurement, the measurement update equations compute *a posteriori* estimate. [47].

Equations 4.1 and 4.2 are from the time update phase. Equation 4.1 projects the state (x_k^-) and equation 4.2 projectsthe covariance (P_k^-) from time step k-1 to step k.

$$\widehat{x_k^-} = A\widehat{x}_{k-1} + Bu_{k-1} \tag{4.1}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{4.2}$$

Equations 4.3, 4.4 and 4.5 belong to the measurement phase. The first step during the measurement update is to compute the Kalman gain (K_k) with Equation 4.3. After obtaining a new measure (Z_k), it is generated an *a posteriori* state estimate as in equation 4.4. The final step is to estimate an *a posteriori* error covariance with equation 4.5.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{4.3}$$

$$\widehat{x}_k = \widehat{x_k^-} + K_k (Z_k - H\widehat{x_k^-}) \tag{4.4}$$

$$P_k = (I - K_k H) P_k^- \tag{4.5}$$

The hand of the user is characterized by the coordinates on 3D space and its orientation. The state to the Kalman filter can be described with four variables (x,y,z and theta) assuming a 4D state space.

$$x_k = \begin{bmatrix} posx_k \\ posy_k \\ posz_k \\ \theta_k \end{bmatrix}, \tag{4.6}$$

In our implementation of the Kalman filter, we assumed that the hand will try to stay close of its last position. For that, we considered a constant position model where the next state is defined by the previous state.

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.7}$$

In the measurement update, in order to not confuse the hand with another object that could appear on the scene, if the segmentation was not efficient, it is assumed that the closest object to the estimated position of the hand is the hand in the actual frame to be the measurement state z_k. This choice is represented in Figure 4.12.

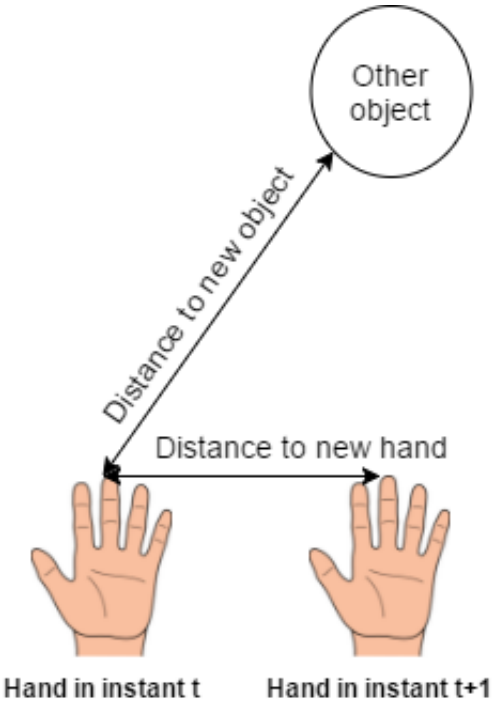


Figure 4.12: Selection of the hand comparing measuring the distance from the hand of the previous frame.

The next figures show the orientation and positions of the controlling hand in the 3 different axis, with the actual measures and the estimated positions computed by the Kalman filter.

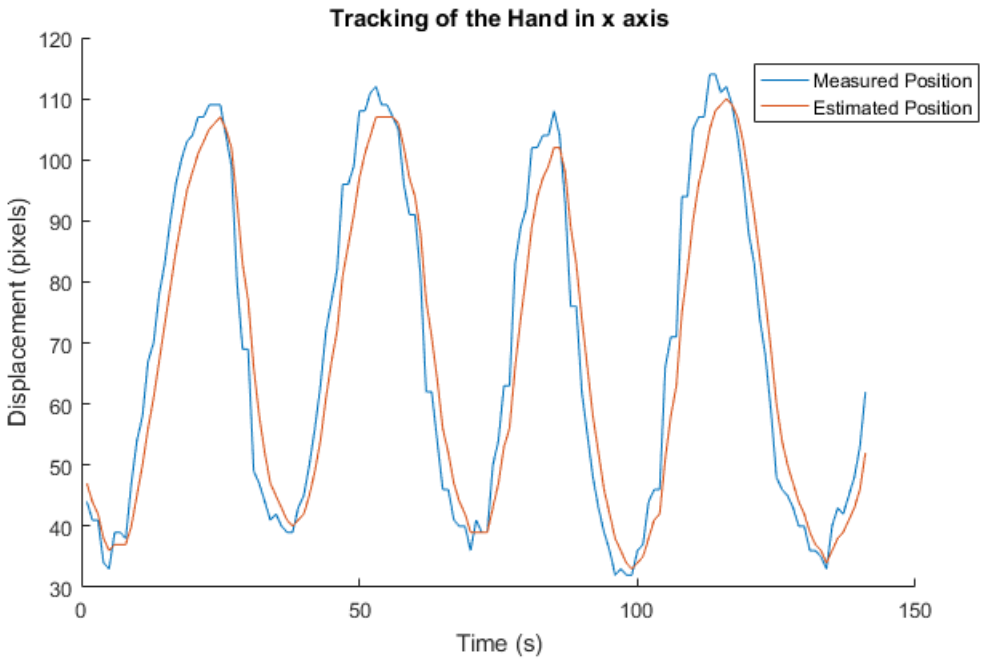


Figure 4.13: X axis hand position variation measured and estimated.

In Figure 4.13, it is presented the position of the hand measured by our system and the estimated position computed by the Kalman Filter in X axis across time.

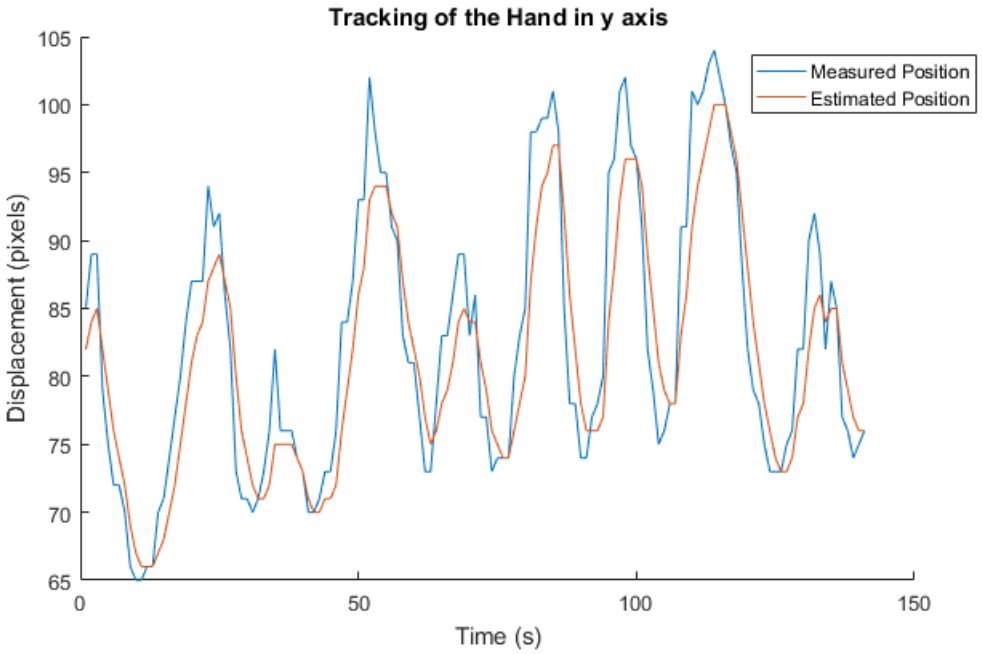


Figure 4.14: Y axis hand position variation measured and estimated.

In Figure 4.14, the measured and estimated positions from the hand in Y axis are presented across time.

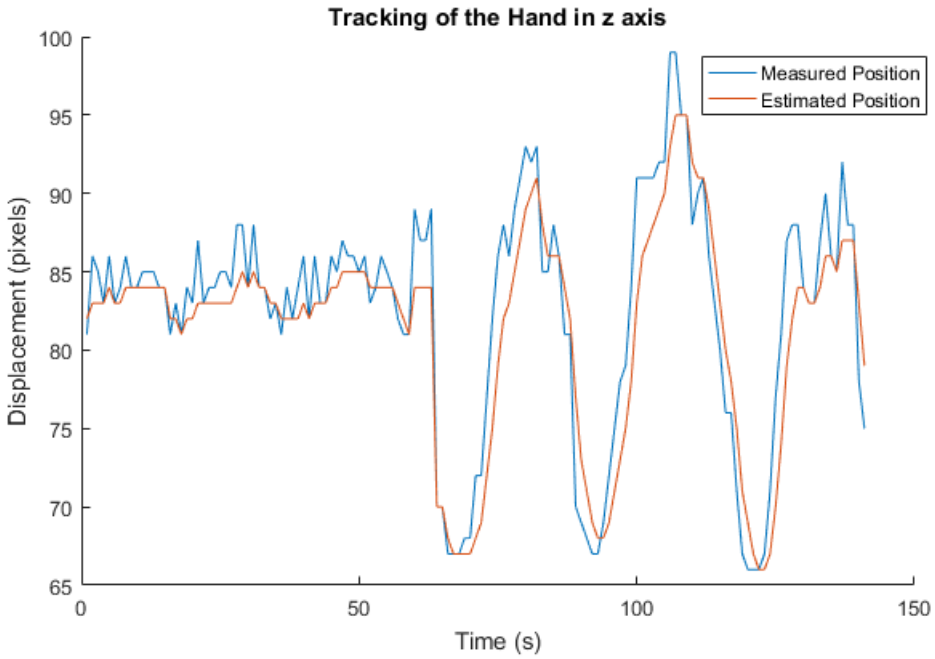


Figure 4.15: Z axis hand position variation measured and estimated.

In Figure 4.15, the measured and estimated positions in the Z axis of the hand are presented.

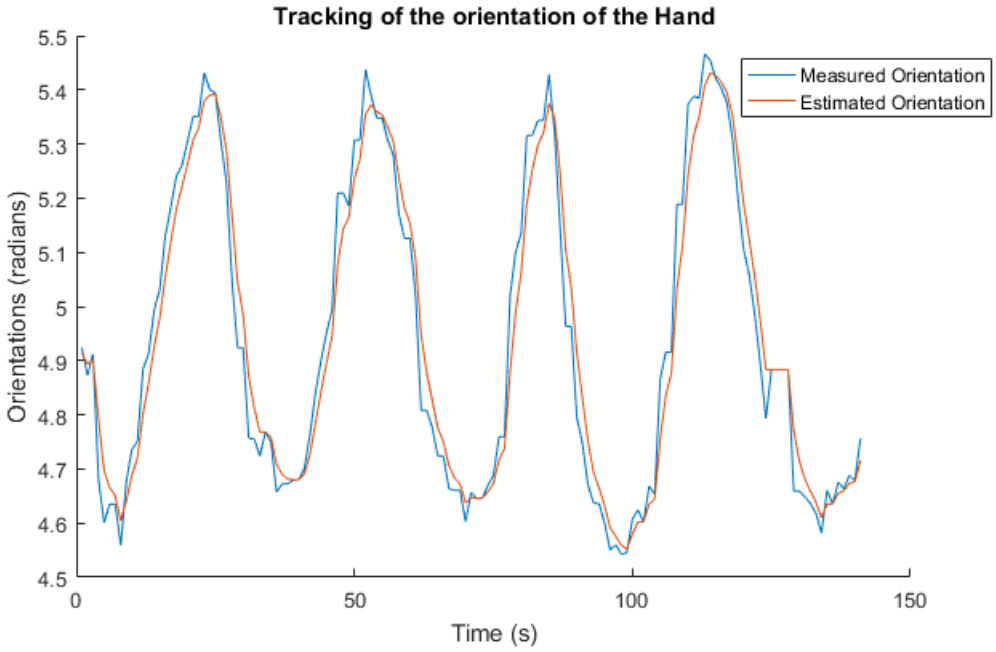


Figure 4.16: Theta orientation variation measured and estimated.

In Figure 4.16, the measured orientation of the arm and its estimation are presented.

In the tracking graphics we observe a smoother trajectory of the hand performed by the Kalman filter. We notice a small delay in our tracking due to the constant position model used, it is caused by the fact that we estimate the next position as the actual position and it will be proportional with the velocity of the hand. Since the gesture is a sequence of positions and for our classification approach we do not analyze the velocity of the hand this delay is not relevant.

In Figure 4.17 the path of the hand in PAUSE gesture is presented.

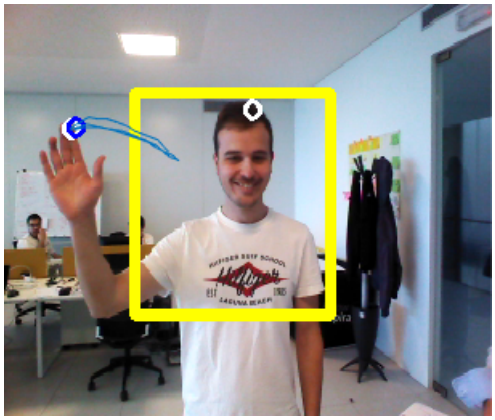


Figure 4.17: Path of the hand in PAUSE gesture. The white circle represents the measured point and the blue the tracked hand.

The state vector obtained with the estimated hand position in each frame allow us to obtain the track to perform the gesture recognition later.

4.7 Classification

In order to identify which gesture was made by the user, a simple Finite State Machine (FSM) for each gesture was implemented.

4.7.1 Finite State Machine

FSM were designed according to the movement of the proposed gestures. Since the hand moves along different coordinate axis (along the z axis for the `START` and along the x axis for the `PAUSE`) or does not even move (in case it is a `STOP` gesture) the proposed approach was designed respecting the gesture parameterization. The states and conditions of the finite state machines were achieved by analyzing the movement of the hand obtained from some demonstration of the gestures.

In figure 4.18, the finite state machine implemented for the `STOP` gesture is presented. In the initial state the hand is moving freely in the "Hand moving freely" state and when it stops moving, it goes to the "No hand movement" state. If it moves again, it will return to the "Hand moving freely". However, if after 1.5 seconds it does not move, it goes to the "Gesture STOP" state were the gesture `STOP` is recognized and a command is sent to the robot.

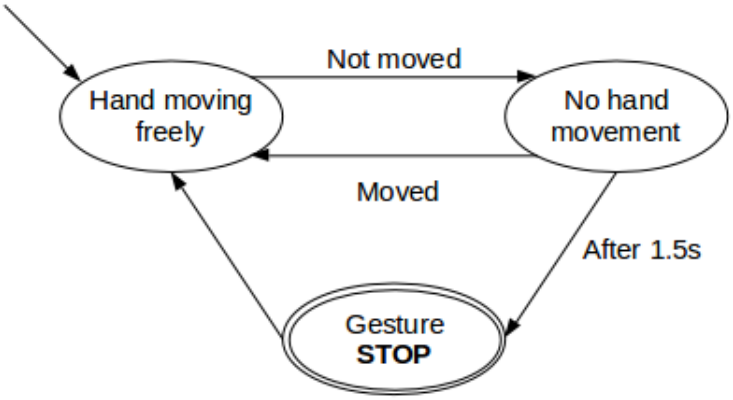


Figure 4.18: Finite state machine of the STOP command.

The FSM for the the `START` gesture is presented in figure 4.19. It starts in "Hand moving freely" and when the hand starts moving back it goes to the "Moving back" state. After this movement, if it starts moving in front it goes to the "Moving front" state, otherwise, if it moves in another direction it returns to the "Hand moving freely". Then, if the two last transitions are repeated and if so, a gesture `START` is detected and sent to the robot. The variation of the hand position is performed mostly in the Z axis.

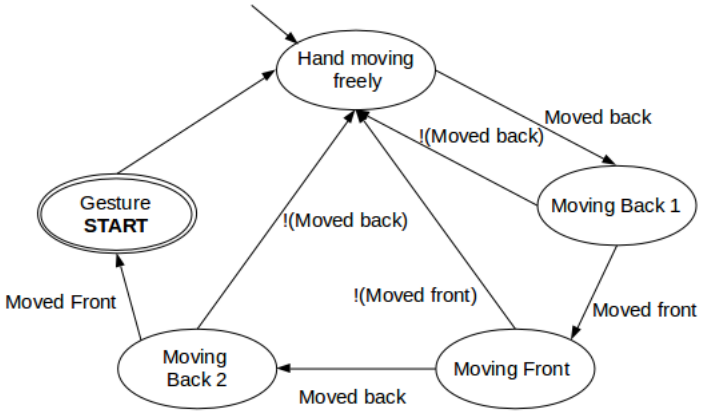


Figure 4.19: Finite state machine of the START command.

The FSM for the PAUSE gesture is similar to the one from START gesture. The difference is the directions of movement, since the direction for the START are front and back, the directions for the PAUSE are right and left. Taking into account this variation it is looked to the X axis instead of the Z axis. The finite state machine for the PAUSE is presented in the figure 4.20.

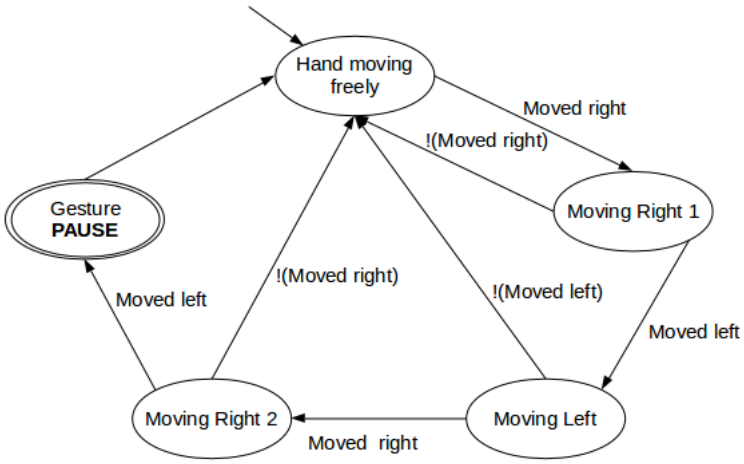


Figure 4.20: Finite state machine of the PAUSE command.

Given the implemented finite state machines, our gesture recognition system was implemented and tested in a real case scenario. Next, we present the results obtained for the proposed approach.

4.8 Experimental results

To evaluate the efficiency of our method, 13 volunteers were asked to perform the gestures. After explaining how to perform the gestures, the volunteers performed them 3 times alternating between gestures.

Figures 4.21 and 4.22 represents a Time-lapse and the hand displacement with the correspondent state of the FSM across time respectively. In the Figure 4.22 we can see that the major

movement is on the X axis having some variation in the Y too. The variation on the Z axis is not significant. The hand is moving freely in the beginning and when the movement of the hand inverts its direction in the X axis where it goes to the "Moving right 1" state, in the next two changes of direction it passes trough the states "Moving Left" and "Moving Right 2" and finally, in the last change of direction the gesture PAUSE is detected like it was parameterized.

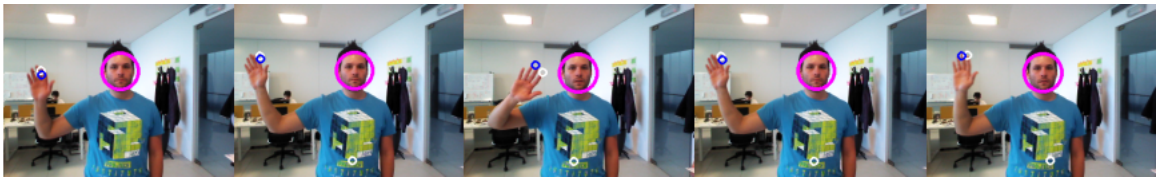


Figure 4.21: Time-lapse of the PAUSE gesture performed across time.

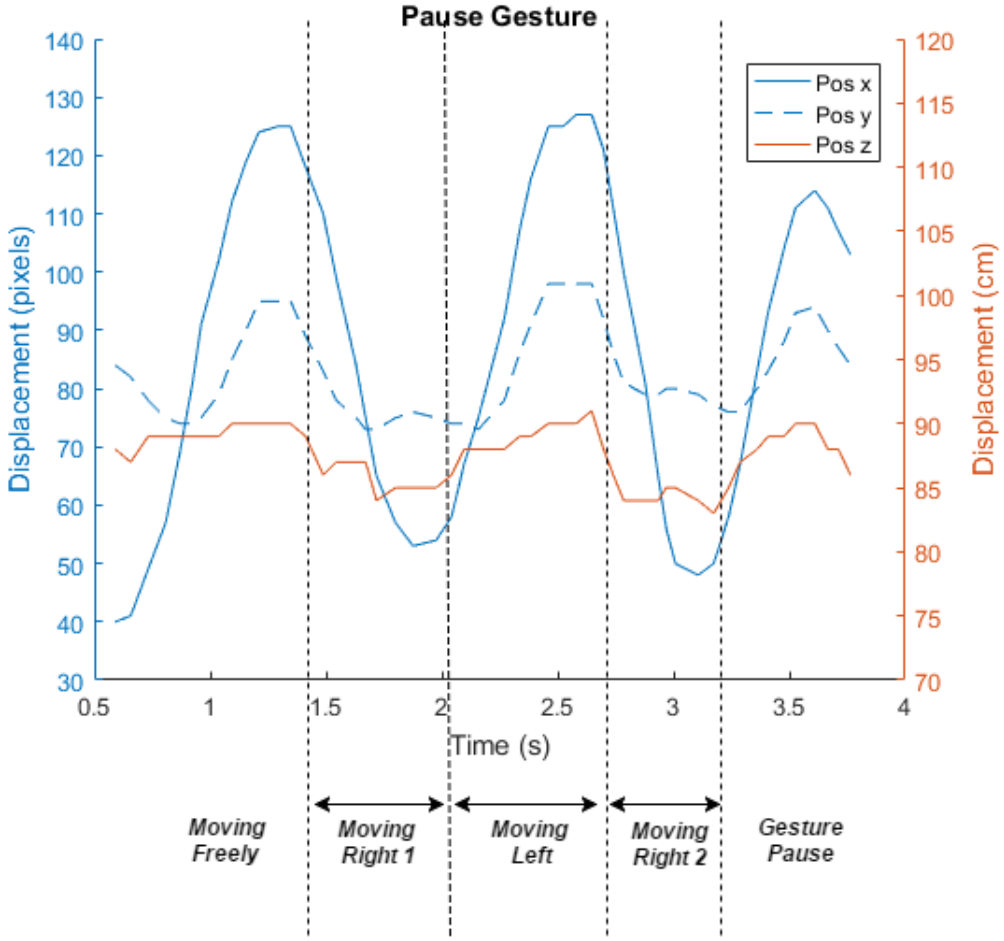


Figure 4.22: Hand displacement in the PAUSE command across time with the correspondent state.

A Time-lapse of the START gesture is represented in Figure 4.23 and its hand displacement with the correspondent state of the FSM is presented in Figure 4.24. Similar to PAUSE gesture, it has a major movement in one axis which in this case is along the Z axis. After passing by the states "Moving Back 1", "Moving Front" and "Moving Back 2", the gesture START is recognized.

Figure 4.23: Time-lapse of the START gesture performed across time.

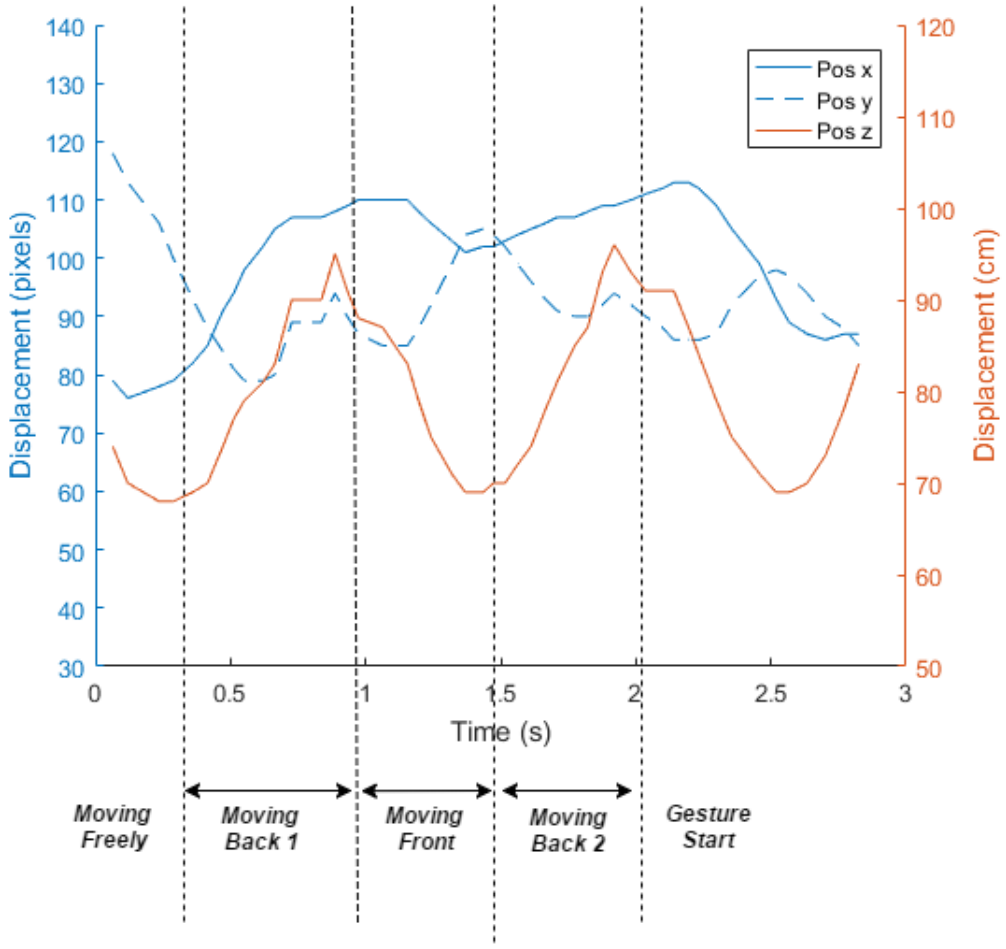


Figure 4.24: Hand displacement in the START command across time with the correspondent state.

The gesture STOP was the third gesture moduled. In Figure 4.25 is represented a Time-lapse of the gesture and in Figure 4.26 the hand displacement and the respective state of the FSM. The hand is moving freely in the beginning and when the movement stops it enters in the "No hand movement" state. After 1.5 seconds the STOP gesture is recognized.

Figure 4.25: Time-lapse of the STOP gesture performed across time.

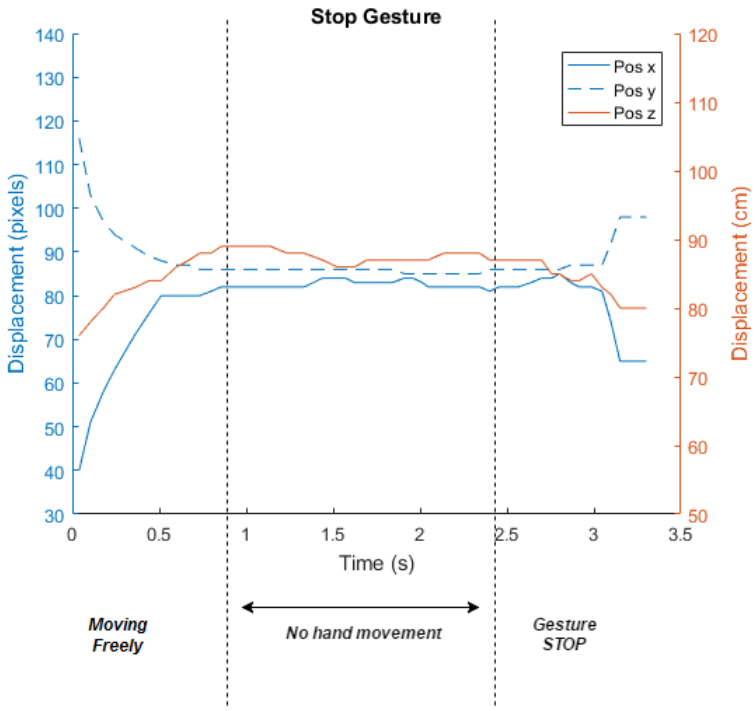


Figure 4.26: Hand displacement in the STOP command across time with the correspondent state.

The results for the gesture recognition were obtained by 13 volunteers and are presented in table 4.1.

Table 4.1: Results of the experimental performances

Gesture	True Positive	False Negative	False Positive
PAUSE	33	6	1
STOP	38	1	1
START	34	5	2

In our results we obtained a certain number of false positives. The false positives are less important when we take into account the state of the robot. Since the START gesture can be only performed when the Robot is stopped and the PAUSE and STOP gesture when it is in use. So we can assume that the detected false positives are irrelevant.

Table 4.2: Correct rate of identification for each gesture.

Gesture	**Accuracy(%)**
PAUSE	84,6
STOP	97,4
START	87,2
Average	89,7

As presented on the table 4.2, for each gesture performed it was achieved a correct rate of 84,6% for the PAUSE, 97,4% for the STOP and 87,2% for the START, with a global accuracy of 89,7%.

Chapter 5

Conclusions and Future work

The proposed solution defines a new approach for Human-Robot Interaction based on the recognition of dynamic gestures to be used in a Service Robot.

Based on the depth image, the face detections result and the user identification received from the robot, a depth based approach was implemented that does not need any kind of calibration and can work in a real case scenario, to perform gesture recognition. Three single-arm dynamic gestures were parameterized for the commands START, STOP and PAUSE in order with the aim of being simple and intuitive for the user.

The developed methodology makes use of the previous described information and for each depth image received, if the face is detected, it performs the following process: The background is removed using the distance of the user to the RGB-D sensor as a threshold value. This distance was computed based on an histogram approach. Then a morphological close and a region growing operation are used in order to remove all the possible noise present on the image. Another threshold using the same distance value is then performed in the original depth image in order to obtain the arms of the user and other possible objects close to the camera. An AND operation is performed with the two resultant images and the original depth image and the arms of the user are obtained. After the segmentation process and with the arms obtained, a PCA is computed to each individual object to detect its orientation and the tip of the hand that is tracked later. The tracking is performed using a Kalman filter using a constant position model that proved to be efficient. In order to identify the three gestures parameterized three simple FSMs were implemented since the gestures are very distinct and not difficult to model. The FSM were low compute expensive and of easy implementation.

The presented solution was tested by several people in a real case scenario, where a real robot was controlled by the user only with gestures. A global accuracy of 89.7% was achieved which indicates the robustness of our proposed approach. Individually, the STOP gesture was recognized with a correct rate of 97.4%, the PAUSE gesture obtained a correct rate of 84.6% and finally the START obtained 87.2%. Regarding the application of our solution on wGO, it clearly contributes for a more natural interface between the robot and supermarket customers.

Even though there is still room for some improvements. At this moment, the system only

works for the right hand but this can easily be replicated for the left hand. Further work in this area could also include the expansion of the work space of the hand, enlarging it so that gestures could be detected in a wider area, or improving the tracking component in order to deal with occlusions. Moreover, it may also be interesting to implement a probabilistic classification method like the Hidden Markov Models in order to improve the accuracy of the system as it will allow us to add more complex gestures to the Human-Robot interface. To finish the integration procedure it will be necessary for the robot to have a PAUSE mode we can use the corresponding gesture commands sent by the described topic in Chapter 3.4.The implementation of the figured interface would also improve the interaction with the user, giving him a visual feedback of his hands performing the gestures.

References

[1] iRobot. Roomba. http://www.irobot.com/For-the-Home/Vacuuming/Roomba.aspx, retrieved April 2017.

[2] Lowe's innovation labs. Innovation robots. http://www.lowesinnovationlabs.com/innovation-robots/, retrieved April 2017.

[3] FollowInspiration. Followinspiration. http://www.followinspiration.pt/, retrieved April 2017.

[4] Blue Frog Robotics. Buddy. http://www.bluefrogrobotics.com/en/home/, retrieved May 2017.

[5] Fraunhofer. Care-o-bot. http://www.care-o-bot-4.de/, retrieved May 2017.

[6] Sanbot. Sanbot s1. http://en.sanbot.com/newsPro/design.html, retrieved May 2017.

[7] Cobalt Robotics. Cobalt. https://www.cobaltrobotics.com/, retrieved May 2017.

[8] MFB van der Burgh, JJM Lunenburg, RPW Appeldoorn, RWJ Wijnands, TTG Clephas, MJJ Baeten, LLAM van Beek, RA Ottervanger, HWAM van Rooy, and MJG van de Molengraft. Tech united eindhoven@ home 2017 team description paper. *University of Technology Eindhoven*, 2017.

[9] Spencer. http://www.spencer.eu/, retrieved May 2017.

[10] J. H. Kim, N. D. Thang, and T. S. Kim. 3-d hand motion tracking and gesture recognition using a data glove. In *2009 IEEE International Symposium on Industrial Electronics*, pages 1013–1018, July 2009. doi:10.1109/ISIE.2009.5221998.

[11] Antonis A. Argyros and Manolis I. A. Lourakis. *Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera*, pages 368–379. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[12] Kui Liu and Nasser Kehtarnavaz. Real-time robust vision-based hand gesture recognition using stereo images. *J. Real-Time Image Process.*, 11(1):201–209, January 2016. URL: http://dx.doi.org/10.1007/s11554-013-0333-6, doi:10.1007/s11554-013-0333-6.

[13] Tudor Ioan Cerlinca and Stefan Gheorghe Pentiuc. *Robust 3D Hand Detection for Gestures Recognition*, pages 259–264. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[14] C. P. Chen, Y. T. Chen, P. H. Lee, Y. P. Tsai, and S. Lei. Real-time hand tracking on depth images. In *2011 Visual Communications and Image Processing (VCIP)*, pages 1–4, Nov 2011. doi:10.1109/VCIP.2011.6115983.

[15] M. Van den Bergh and L. Van Gool. Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 66–72, Jan 2011. doi:10.1109/WACV.2011.5711485.

[16] Archana Ghotkar, Pujashree Vidap, and Kshitish Deo. Dynamic hand gesture recognition using hidden markov model by microsoft kinect sensor. *International Journal of Computer Applications*, 150(5):5–9, Sep 2016. URL: http://www.ijcaonline.org/archives/volume150/number5/26087-2016911498, doi:10.5120/ijca2016911498.

[17] Chang-Beom Park and Seong-Whan Lee. Real-time 3d pointing gesture recognition for mobile robots with cascade {HMM} and particle filter. *Image and Vision Computing*, 29(1):51 – 63, 2011. URL: http://www.sciencedirect.com/science/article/pii/S0262885610001149, doi:https://doi.org/10.1016/j.imavis.2010.08.006.

[18] Pavel Senin. Dynamic Time Warping Algorithm Review. Technical Report CSDL-08-04, Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, December 2008. URL: http://csdl.ics.hawaii.edu/techreports/08-04/08-04.pdf.

[19] A. Ramey, V. Gonzalez-Pacheco, and M. A. Salichs. Integration of a low-cost RGB-D sensor in a social robot for gesture recognition. In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 229–230, March 2011. doi:10.1145/1957656.1957745.

[20] Orbbec. Orbbec astra. https://orbbec3d.com/product-astra/, retrieved April 2017.

[21] Robot operating system. http://wiki.ros.org/, retrieved March 2017.

[22] International Federation of Robotics. Service robots. https://ifr.org/service-robots/, retrieved April 2017.

[23] António Neves Tiago Esteves Patrick de Sousa, Luís Texeira. Human-robot interaction based on gestures for service robots. VipIMAGE, 2017.

[24] Kris Doelling, Jeongsik Shin, and Dan O. Popa. Service robotics for the home: A state of the art review. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '14, pages 35:1–35:8, New York, NY, USA, 2014. ACM. URL: http://doi.acm.org/10.1145/2674396.2674459, doi:10.1145/2674396.2674459.

[25] RobotCup. Robotcup@home. http://www.robocupathome.org/about, retrieved May 2017.

[26] M. Malfaz M.A. Salichs V. Gonzalez Pacheco, A. Castro-Gonzalez. Human robot interaction in the monarch project. pages –, 2015.

[27] David Feil-Seifer and Maja J. Matarić. *Human RobotHuman–robot interaction (HRI) InteractionInteraction human robot*, pages 4643–4659. Springer New York, New York, NY, 2009. URL: http://dx.doi.org/10.1007/978-0-387-30440-3_274, doi:10.1007/978-0-387-30440-3_274.

[28] Robert B. Burns and SpringerLink (Online service). *Verbal and Non-verbal Communication*, pages 223–235. Dordrecht : Springer Netherlands, 1991. URL: http://dx.doi.org/10.1007/978-0-585-30665-0.

[29] Nikolaos Mavridis. A review of verbal and non-verbal human robot interactive communication. *Robotics and Autonomous Systems*, 63:22 – 35, 2015. URL: http://www.sciencedirect.com/science/article/pii/S0921889014002164, doi:http://dx.doi.org/10.1016/j.robot.2014.09.031.

[30] Veton Këpuska and Gamal Bohouta. Comparing speech recognition systems (microsoft api, google api and cmu sphinx).

[31] Fatik Baran Mandal. Nonverbal communication in humans. *Journal of Human Behavior in the Social Environment*, 24(4):417–421, 2014. URL: http://dx.doi.org/10.1080/10911359.2013.831288, arXiv:http://dx.doi.org/10.1080/10911359.2013.831288, doi:10.1080/10911359.2013.831288.

[32] Siddharth S. Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, 2015.

[33] J. S. Sonkusare, N. B. Chopade, R. Sor, and S. L. Tade. A review on hand gesture recognition system. In *2015 International Conference on Computing Communication Control and Automation*, pages 790–794, Feb 2015. doi:10.1109/ICCUBEA.2015.158.

[34] Shamir Alavi, Dennis Arsenault, and Anthony Whitehead. Quaternion-based gesture recognition using wireless wearable motion capture sensors. *Sensors*, 16(5), 2016. URL: http://www.mdpi.com/1424-8220/16/5/605, doi:10.3390/s16050605.

[35] Nazrul H. Adnan, Khairunizam Wan, A.B. Shahriman, S.K Zaaba, Shafriza nisha Basah, Zuradzman M. Razlan, D. Hazry, M. Nasir Ayob, M.Nor Rudzuan, and Azri A. Aziz. Measurement of the flexible bending force of the index and middle fingers for virtual interaction. *Procedia Engineering*, 41:388 – 394, 2012. URL: http://www.sciencedirect.com/science/article/pii/S1877705812025891, doi:http://dx.doi.org/10.1016/j.proeng.2012.07.189.

[36] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma. A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. *IEEE Sensors Journal*, 16(16):6425–6432, Aug 2016. doi:10.1109/JSEN.2016.2581023.

[37] Kunal R Jambhulkar. Review on Sensor based Hand Gesture Recognition System. *International Journal of Research in Engineering & Advanced Technology*, 5(1):33–36, 2017.

[38] Cristina Manresa, Javier Varona, Ramon Mas, and Francisco Perales. Hand tracking and gesture recognition for human-computer interaction. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 5(3):96–104, 2005. URL: http://elcvia.cvc.uab.es/article/view/109.

[39] Sangheon Park, Sunjin Yu, Joongrock Kim, Sungjin Kim, and Sangyoun Lee. 3D hand tracking using kalman filter in depth space. *EURASIP Journal on Advances in Signal Processing*, 2012(1):36, 2012.

[40] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. URL: http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb, doi:10.1023/B:VISI.0000013087.49260.fb.

[41] Sergio Rodriguez, Artzai Picon, and Aritz Villodas. Robust vision-based hand tracking using single camera for ubiquitous 3d gesture interaction. In *Proceedings of the 2010 IEEE Symposium on 3D User Interfaces*, 3DUI '10, pages 135–136, Washington, DC, USA, 2010. IEEE Computer Society. URL: http://dx.doi.org/10.1109/3DUI.2010.5444702, doi:10.1109/3DUI.2010.5444702.

[42] Openni. Nite. http://openni.ru/files/nite/, retrieved April 2017.

[43] Microsoft. Kinect. https://developer.microsoft.com/pt-pt/windows/kinect, retrieved April 2017.

[44] C. Bellmore, R. Ptucha, and A. Savakis. Interactive display using depth and rgb sensors for face and gesture control. In *2011 Western New York Image Processing Workshop*, pages 1–4, Nov 2011. doi:10.1109/WNYIPW.2011.6122883.

[45] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[46] John G. Allen, Richard Y. D. Xu, and Jesse S. Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In *Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing*, VIP '05, pages 3–7, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc. URL: http://dl.acm.org/citation.cfm?id=1082121.1082122.

[47] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill, NC, USA, 1995.

[48] Michael Isard and Andrew Blake. Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. URL: http://dx.doi.org/10.1023/A:1008078328650, doi:10.1023/A:1008078328650.

[49] Michael Isard and Andrew Blake. *A smoothing filter for condensation*, pages 767–781. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. URL: http://dx.doi.org/10.1007/BFb0055703, doi:10.1007/BFb0055703.

[50] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989. doi:10.1109/5.18626.

[51] Zoubin Ghahramani. Hidden markov models. chapter An Introduction to Hidden Markov Models and Bayesian Networks, pages 9–42. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2002. URL: http://dl.acm.org/citation.cfm?id=505741.505743.

[52] C. Yang, Yujeong Jang, J. Beh, D. Han, and H. Ko. Gesture recognition using depth-based hand tracking for contactless controller application. In *2012 IEEE International Conference on Consumer Electronics (ICCE)*, pages 297–298, Jan 2012. doi:10.1109/ICCE.2012.6161876.

[53] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4207–4215, June 2016. doi:10.1109/CVPR.2016.456.

[54] Giovanna Sansoni, Marco Trebeschi, and Franco Docchio. State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors*, 9(1):568–601, 2009. URL: http://www.mdpi.com/1424-8220/9/1/568, doi:10.3390/s90100568.

[55] O. Bilaniuk, E. Fazl-Ersi, R. Laganière, C. Xu, D. Laroche, and C. Moulder. Fast lbp face detection on low-power simd architectures. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 630–636, June 2014. doi:10.1109/CVPRW.2014.96.

[56] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[57] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[58] Hervé Abdi and Lynne J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010. URL: http://dx.doi.org/10.1002/wics.101, doi:10.1002/wics.101.